ANALYZING HEALTH-RELATED BEHAVIORS USING FIRST-PERSON VISION

A Thesis Proposal
Presented to
The Academic Faculty

By

Yun Zhang

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology

August 2021

ANALYZING HEALTH-RELATED BEHAVIORS USING FIRST-PERSON VISION

Approved by:

Dr. James M. Rehg, Advisor
School of Interactive Computing
Georgia Institute of Technology

Dr. Mark A. Clements, Co-Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Omer T. Inan
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Gregory D. Abowd
Department of Electrical and Computer Engineering
Northeastern Univeristy

Dr. Thomas Ploetz
School of Interactive Computing
Georgia Institute of Technology

Dr. Nabil Alshurafa
Department of Preventive Medicine and Department of Computer Science
Northwestern University

Date Approved: May 21, 2021

This thesis is dedicated to my Mom, who brightens up my life with her support, trust and constant love.

ACKNOWLEDGEMENTS

First I am grateful to my advisor, Prof. James M. Rehg, for showing me the exciting research area of computer vision and for his invaluable guidance, continuous support, and patience during my PhD study. He helped me grow as a professional researcher with his encouragement when I got stuck, his tolerance when we had disagreements, and his direction when I made mistakes.

I would like to thank the rest of my thesis committee: Prof. Mark A. Clements, Prof. Omer T. Inan, Prof. Gregory D. Abowd, Prof. Thomas Ploetz and Prof. Nabil Alshurafa for their insightful comments and questions. Their constructive suggestions greatly contributed to my research.

My appreciation also goes out to my collaborators and colleagues for the time spent together on the technical discussions and social conversations. Their precious help and support have improved my study and life in the United States.

Finally, I would like to express my gratitude to my family, especially my mom. Without their tremendous understanding and encouragement, it would have been impossible for me to complete my study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Wearable sensors are increasingly affordable and easy to deploy, and as a result, they are widely-used in mobile health applications. Sensors such as accelerometers and gyroscopes are broadly-used due to their low cost, small size, and relatively low power consumption. More recently, it has become feasible to deploy wearable cameras in mobile health applications with two primary benefits: 1) when cameras are head-mounted, they can capture the health-related visual attention of the participant, which is difficult to measure via other sensing modalities; 2) cameras can capture health-related behaviors under a wide range of conditions and contexts. When cameras are combined with other wearable sensors, the video can be used to curate labeled examples of behaviors under field conditions and thereby drive the development of machine learning-based models, provided that the sensor data is time-synchronized.

The goal of this thesis is to develop methods for analyzing video from wearable cameras, an area known as egocentric vision, that can enable both automatically-derived measures of behavior and the automatic synchronization of video with other sensing modalities in order to facilitate the large scale collection of labelled training data for use in building machine learning models. I begin by describing a model for egocentric action recognition which leverages the shared motion and appearance properties of different actions to enable zero shot learning (the model can predict novel actions that it has not been trained on.) Second, I develop a method to analyze video from a head-worn camera and quantify the participant's attention to screens (*e.g.*, monitors, smartphones), without using an eye-tracker. Finally, I present a method based on a weighted kernel density estimation approach to automatically synchronize the timestamps of a wearable camera and wearable accelerometer. The method is able to estimate the time offset between multiple modalities of sensor data collected from devices mounted on different locations in the natural field environment without introducing additional participant burden.

CHAPTER 1
INTRODUCTION

Human behavior is fundamentally related to health. As a consequence of advances in biomedical science over the past 100 years, the major cause of death and the major sources of cost in health-care are no longer microbial agents or accidents. Instead, they are health conditions that are derived from human behavior, which are frequently chronic and recurring. 38% of deaths are caused by poor diet, physical inactivity, alcohol and tobacco [1]. A significant source of cost in United States healthcare system is derived from chronic health conditions resulting from the consequences of smoking, physical inactivity, poor diet, and substance abuse, among many other health-related behaviors.

As a consequence of the growing health burden of chronic conditions, there has been an increasing focus on preventative medicine and behavior change as a means to target major cost drivers such as cancer and heart disease. The public health goal of reducing or eliminating unhealthy behaviors such as smoking is often aligned with the goals and desires of the affected individuals. Many people desire to improve their long-term health outcomes by losing weight, becoming more physically active, and quitting smoking. However, bringing about meaningful long-term behavior change is extremely difficult. For example, 55.4% of smokers tried quitting in 2015, but 7.4% succeeded [2]. Moreover, abstinence is often associated with unhealthy eating behaviors, such as snacking, with the result that 80% of ex-smokers experience weight gain [3]. An important goal of mobile health research is developing more effective tools that enable individuals to be more successful in bringing about positive behavior change.

A key goal is to identify and deliver effective, personalized and scalable approaches to behavior change. A significant opportunity enabled by wearable devices is the possibility of measuring, under naturalistic conditions, the physiological state, context, and

environmental exposures that are correlated with health-related behaviors. Specifically, we can recognize the moments when participants are engaged in health-related behaviors and identify their antecedents. The ability to identify the relationship between risk factors and adverse outcomes in daily life situations could enable new approaches to designing and providing behavioral health interventions.

The starting point for our work is the observation that wearable cameras can play an important role in enabling the development of new mobile health models and interventions with the potential to affect behavior change. This is because the first-person video (FPV) recorded by a wearable camera provides a unique vantage point for inferring both the context of a participant (the environment and situation that they are in) as well as the visual inputs they receive along with their experiences. The camera motion implicitly encodes the camera wearer's attention and intention. Together with the image, it contains egocentric cues for understanding objects and activities. Furthermore, wearable cameras are increasingly available, due to the success of products like GoPro and broad interest in augmented reality applications. Not only are people increasingly willing to wear cameras and capture aspects of their daily life, they are also sharing the resulting videos over social media. In fact, a recent study demonstrated that a significant fraction of socially-driven micro-videos include egocentric viewpoints [4]. While cameras can play a useful role in measuring health-related behaviors, a complete solution will ultimately require multiple sensing modalities. This is due to many factors, including limited battery life, which precludes continuous daily monitoring with wearable cameras, concerns about privacy and secondary capture, and limited camera field of view, which necessitates the use of other sensors to capture the full range of behaviors. In this context, wearable cameras can be a useful tool for curating and labeling examples of behaviors captured under field conditions using other sensors such as accelerometers, gyroscopes, and so forth.

One area in which wearable cameras are widely-used, including in combination with other wearable sensing modalities, is in the domain of eating detection [5, 6]. While the arm

movements associated with bringing food to the mouth can be sensed via an accelerometer, for example, the curation of training datasets for model development requires a way to determine whether food is present, and this can be accomplished using a wearable camera. Despite the popularity of work in this area, the broader usage of wearable cameras in mobile health applications is under-explored. This raises two questions that this dissertation addresses: First, how can the facility of egocentric video to measure aspects of the user's visual attention be leveraged to sense health-related behaviors? Second, how can we enable the automatic synchronization of a wearable camera with other sensing modalities in order to facilitate the large-scale use of wearable camera video for dataset curation and labeling?

1.1 Thesis Statement

Wearable cameras capture the participant's attention and context, enabling behavior measurement tasks like quantifying screen use and facilitating the labeling of in-the-wild sensor data from modalities such as accelerometry via automated synchronization with video.

1.2 Overview

This dissertation is organized into three main topics: first-person action decomposition and zero-shot learning (chapter 3), screen-use activity detection and associated attention estimation (chapter 4), and the automated synchronization of video and accelerometry signal streams (chapter 5). An overview of each these topics follows.

1.2.1 First-person Action Decomposition and Zero-shot Learning

This chapter presents a novel zero-shot learning approach to first-person action recognition. In this work, we decompose a first-person action into verb and noun. We then study how the coupling of an action's constituent verb and noun affects the learners' ability to learn them separately and to combine them to perform recognition. We compare different information fusion methods on conventional action recognition and zero-shot learning,

of which the latter is a strong indication of the feature's ability to capture one concept (verb/noun) and not be confounded by the other. To achieve the decoupling of verb/noun concepts, we extract features that are specialized for each of them. Specifically, we use improved dense trajectories and convolutional neural network activations. We show that by constructing specialized features for the decomposed concepts, our method succeeds in zero-shot learning. More surprisingly, it also outperforms results in conventional activity recognition when the performance gaps of different features on verb/noun concepts are significant. Prior work has demonstrated the ability to recognize cooking actions from egocentric video, which suggests that automated video analysis can be useful in monitoring behaviors related to feeding. My work on zero-shot action learning demonstrates the feasibility of a scalable learning approach, in which novel action categories can be added without the need to collect extensive training data.

1.2.2 Screen-use Activity Detection and Associated Attention Estimation

Studies have linked excessive TV watching to obesity in adults and children. In addition, TV content represents an important source of visual exposure to cues which can effect a broad set of health-related behaviors. This chapter presents a novel wearable sensing system which can detect moments of screen-watching during daily life activities. Machine learning techniques are used to analyze video captured by a head-mounted wearable camera. Although wearable cameras do not directly provide a measure of visual attention, we show that attention to screens can be reliably inferred by detecting and tracking the location of screens within the camera's field-of-view. Utilizing a computational model of the head movements associated with attention to screens, my method can identify and quantify screen-watching events. We have evaluated our method on TV watching videos recorded from 16 participants in a home environment as well as footage of screen-watching from mobile devices. Our model achieves a precision of 0.917 and a recall of 0.945 in identifying attention to screens. The third-person annotations used to determine accuracy were

validated and our system was further evaluated in a multi-screen environment using gold standard attention measurements obtained from a wearable eye-tracker. Finally, wearable cameras were deployed in a naturalistic study in which participants captured video during unconstrained daily life activities, which included significant bouts of screen watching. The system achieved a precision of 0.87 and a recall of 0.82 on challenging naturalistic videos capturing the daily life activities of participants.

1.2.3 Synchronization of Video and Accelerometry

The development and validation of computational models to detect behaviors of daily living (e.g., eating, smoking, brushing) using wearable devices requires labeled data collected from the natural field environment with tight time synchronization of the micro-behaviors (e.g., start/end times of hand-to-mouth gestures during a smoking puff or an eating gesture) and the associated labels. Video data is increasingly being used for such label collection. Unfortunately, wearable devices and video cameras with independent (and drifting) clocks make tight time synchronization challenging. To address this issue, I present the Window Induced Shift Estimation method for Synchronization (SyncWISE) approach. The feasibility and effectiveness of the method was demonstrated by synchronizing the timestamps of a wearable camera and wearable accelerometer from 163 videos representing 45.2 hours of data from 21 participants enrolled in a real-world smoking cessation study. The novel approach shows significant improvement over the state-of-the-art method, even in the presence of high data loss, achieving 90% synchronization accuracy given a synchronization tolerance of 700 milliseconds. This method also achieves state-of-the-art synchronization performance on the CMU-MMAC dataset.

My dissertation makes the following contributions:

- A method for recognizing first-person actions that can be described by combinations of verbs and nouns [7].
- A method for detecting screen-use moments and localizing the screen being focused

on from first-person videos [8].

- An approach to synchronizing first-person videos and acceleration signals [9].
- The collection and analysis of first-person vision datasets involving screen use and smoking activities.

CHAPTER 2
LITERATURE SURVEY

In this chapter, I describe the primary background literature for my thesis work.

2.1 First-person Action Decomposition and Zero-shot Learning

2.1.1 Action Recognition with Context

Actions can be characterized by the objects that are involved, the motions that are produced, and the scenes in which they occur. Previous works used this contextual knowledge to perform action recognition. Marszalek *et al*. [10] and Vu *et al*. [11] used the scene context to recognize actions. We focus on human-object interactions, which have been studied in action recognition for videos [12, 13] and images [14]. These previous works model the relationship between objects and actions for recognition. In this work, we focus on actions that are steps in a meal preparation activity captured from the first-person perspective. These actions can be described in terms of the motion of the person (verb) and the objects (noun) he or she interacts with. Our goal is to model noun-verb relationships in object manipulation actions with first-person vision.

2.1.2 First-person Action Recognition

Our work builds on many previous efforts in first-person action recognition [15, 16, 17, 18]. Many of them used hand-object relationships to recognize actions. Prior work [19] has modeled actions as the stage changes of objects. Other work [20], modeled action and gaze location with object- and appearance-based features. Pirsiavash and Ramanan [21] built an active object detector to distinguish the objects that are being manipulated. Ma *et al*. [18] built a two-stream deep learning pipeline to perform action recognition via verb-

noun decomposition. Their approach fused verb and noun features so that the relationships among verbs, nouns and actions are utilized. Park and Shi [22] focused on modeling a specific action pattern corresponding to shared attention in recovering social saliency maps from first person video. McCandless and Grauman [17] used spatio-temporal features to summarize first-person videos. None of these works investigated zero-shot learning. In comparison, we model the verb and noun labels as attributes of an action and use them to achieve zero-shot action recognition.

2.1.3 Zero-shot Learning

Outside of the FPV domain, there have been several works on zero-shot learning in image classification [23, 24, 25, 26, 27, 28] and video action recognition [29]. These methods have not been applied to first-person actions. One way to transfer knowledge from training classes to novel classes is to use an attribute space [30]. Attributes can be modeled by multiple binary classifiers or regressors that estimate the likelihood that each attribute is present. An alternative approach is through an embedding space [31, 32]. Jain *et al*. [33] developed an unsupervised framework for zero-shot action recognition through an object embedding space. The videos and action labels are represented with a object embedding through an image classifier and a word2vec translation.

2.2 Screen Use Detection

Having reviewed the prior works on using a wearable camera to monitor cooking activities and other actions from a first-person viewpoint, we now turn our attention to a novel use of FPV to monitor TV-watching behaviors and review the prior work in that area. While there are a variety of works on detecting users' attention to TV screens or measuring screen time, no prior work has developed an automatic approach to measuring attention to screens based on first-person vision. Prior works can be organized into four broad categories: Use of cameras mounted on screens to monitor watching behaviors, use of wearable sensors to

measure screen proximity, use of wearable eye trackers, and use of wearable cameras with manual annotations. Below we review each of these approaches and their advantages and limitations. In addition, we review prior work on measuring visual attention using wearable cameras and other imaging modalities, as this is a key component of our approach.

2.2.1 On-Device Cameras

Detecting and analyzing TV viewers' watching behavior using a TV-mounted camera has been addressed in several prior works [34, 35]. RGB, depth and infrared cameras have been used to record the audience. Features such as facial expression, head pose, and the magnitude of a viewer's motions can be extracted from videos and used to classify the viewer's attention. Hernandez *et al*. extracted features characterizing facial and head gestures from videos recorded by a camera on the TV, in order to predict the engagement level of TV viewers [35]. Lee *et al*. also built a system to measure the engagement level of children [36]. They used color and depth videos captured with a Kinect sensor to extract the body skeleton and build features for classification. Takahashi *et al*. extracted head pose information from color and depth videos captured by a Kinect sensor to identify whether the viewer was gazing at the TV [37].

Passive, appearance-based approaches to gaze tracking have also been developed for mobile devices. Machine learning algorithms based on deep convolutional neural networks [38] or using hand-crafted features with a random forest classifier [39] have been used for uncalibrated gaze estimation during natural use of tablets or smartphones by means of their front-facing cameras. In related work, Ye *et al*., Chong *et al*., and Smith *et al*. addressed the problem of automatically determining when a user was looking towards an embedded camera via the appearance properties of the eye [40, 41, 42]. These works could be used to determine when subjects are gazing at the screen of a mobile device. However, these works do not address the problem of linking together all of the screen exposures experienced by an individual.

2.2.2 Color Sensors

Fletcher *et al*. [43] proposes an automated system to measure screen time exposure. It makes use of a wearable color sensor and an algorithm that can distinguish between electronic screens and ambient lighting. Since the sensor is located on the wrist, it can detect screen proximity when the sensor is facing the electronic screen. However, the sensor may not be facing the screen or it may become occluded, and in addition, a subject may be near a screen but not attending to it. In contrast, we utilize a head-mounted camera to estimate fine-grained measures of attention to screens.

2.2.3 Eye-Tracking Cameras

Screen-based and glasses-based devices can be used to track the attention of a user to one or more screens. Eye-tracking sensors are quite expensive in comparison to the video recorders used in our work and in the prior work in Section 2.2.4 which is a major barrier to their use in large-scale studies. These glasses are also potentially more stigmatizing (obtrusive), and more burdensome to use as they are tethered to a recording device. This motivates our decision to use video recorded from a forward-facing scene camera mounted in a pair of glasses, in conjunction with machine learning algorithms, to monitor viewers' screen watching activities.

Some work has been done to study TV viewers' visual attention patterns using eye-tracking devices [44, 45]. Cauchard *et al*. [46] studied the effects of visual separation in a single-device-multi-display mobile environment, when the multiple displays are in the same field of view and when they are not, as well as when the device is fixed or mobile. Brown *et al*. [47] and Apoalaza *et al*. [48] discussed the limitations of eye-tracking systems. They investigated the ecological validity of such devices in experiments that explored how people allocated attention while watching TV with companion content on a second device by comparing the results from an eye-tracking experiment with those from an experiment using surveillance cameras, along with manual coding and an object tracking algorithm.

We tested our automated system in a similar multi-device experimental setting.

Note that the accurate spatial measurement of the gaze point which can be obtained from eye tracking is indispensable in studies that examine the fine-grained allocation of attention within a screen. In contrast, our approach cannot accurately determine where the subject is looking inside the screen, but we *can* reliably identify attention to screens without using the eye-facing cameras required for eye tracking.

2.2.4 Wearable Cameras with Manual Annotation

A few other studies have explored the use of wearable cameras to measure TV-watching behavior in free-living settings. In [49], participants wore an Autographer wearable camera attached to the neckline of their shirt when they were at home. Camera images were coded to show whether the television was visible and identify the location of the participant (i.e. which room they were in), or were labeled as un-codeable. In [50], participants wore the SenseCam device on a lanyard around their neck during waking hours for 3–5 days. The device took photos every 10–15 seconds. The images were hand-coded based on a published physical activity compendium [51]. Kerr *et al*. [50] annotated single images and grouped together a series of at least five consecutive images (approximately 2 minutes) with the same behavior label as an "event." Our system eliminates the need for such manual labeling, and our goal is to provide a tool that can be used for large-scale data collection. By capturing video instead of temporally-separated images, we can access motion cues which are valuable in estimating attention, and we can estimate fine-grained onset and offset times. The downside of capturing video is the greatly-reduced battery life.

2.2.5 Visual Attention Estimation

Our approach to monitoring TV watching utilizes the estimation of the subject's gaze target from FPV. Here we briefly review the literature of attention estimation. The question of how visual attention is allocated has attracted significant interest from researchers. Recent

reviews of visual attention research from the computational and neuroscience perspectives are provided by Hayhoe and Ballard [52] and Petersen and Posner [53], respectively. One line of work uses eye-tracking data collected during screen-viewing to model the saliency of image [54, 55] and video [56] locations. Other works have used wearable eye-tracking cameras to study visual attention in natural scenes [57, 58, 59, 60, 61]. Related to our approach, Li *et al*. [61] estimated the point-of-gaze of a participant by analyzing video collected with a front-facing headworn camera and validated their performance using a wearable eye tracker. They focused on activities such as cooking that involve extensive hand-eye coordination, a scenario in which the participant's hands are often visible. They predict the gaze point by using head motion and hand configuration cues. In contrast, TV watching does not involve a significant amount of hand-eye coordination, and the participant's hands are usually not visible in the scenarios that we address.

2.3 Synchronization of Video and Acceleration Signals

The time synchronization of multiple sensor streams is a long-standing challenge that cuts across a broad range of application domains and has a long history, ranging from the invention of the clapperboard in 1931 to synchronize audio and video during filming, to the protocols used to synchronize sensor networks [62]. This review is focused on methods for synchronizing video with wearable sensor streams for mobile sensing applications. We identify four categories of approaches: 1) Naturalistic methods, of which our work is an example, which do not impose any special requirement on signal capture; 2) Explicit methods, which enforce synchronization at the hardware or software level during capture; 3) Participant-based methods, which require specific actions by participants to achieve synchronization; and 4) Manual approaches which rely on human observation of video and other signals to identify synchronization points.

2.3.1 Naturalistic Methods

The goal of these methods is to handle sensor data captured in the field without special hardware or specific participant behaviors. The closest previous work to ours is Fridman *et al*. [63], which describes a cross-correlation-based method designed to synchronize multi-modal signals for research in autonomous driving. Their approach assumes that all moments in time are equally good for synchronizing signals, and they use global cross-correlation to utilize the maximum amount of data. This is effective because their sensors are rigidly mounted and the coordinate axes are aligned and calibrated. In contrast, mobile wearable sensing is plagued by much greater sensor noise (due to sensors being worn improperly), variable alignment between sensor axes, and partial observability, meaning that sensors do not always capture the same phenomena with the result that not all moments in time are equally plausible for synchronization. Our matching approach, which uses windowed cross-correlation in a weighted kernel density estimation framework, addresses partial observability by identifying which windows of data provide reliable signals for synchronization. Our PCA-alignment approach provides a means of automatically aligning the coordinate frame axes across multiple sensors. Our experimental evaluation in Sec. 5.4 demonstrates the benefits of our approach over the baseline method from [63] on two datasets.

A related set of naturalistic methods provide synchronization solutions for GPS navigation systems, of which [64, 65] are representative examples. Skog *et al*. [65] provide a Kalman filter-based solution for clock drift that exploits the fact that both GPS-receiver and IMU provide signals that directly relate to the spatial location of the sensor system. In contrast, in our setting, the optical flow that we compute from the video cannot be directly related to the accelerometry stream due to the partial observability problem.

Another set of related methods addresses the automatic synchronization of multiple *video streams* [66, 67, 68]. These approaches leverage the fact that video is a single modality with unique spatiotemporal properties. In contrast, our work addresses the case of syn-

chronizing across sensor modalities, which requires the extraction of an appropriate feature representation from each sensor's signal. Related work by Chung and Zisserman [69] uses deep learned representations to align audio and video streams in the context of correcting lipsynch effects in video dubbing. Their solution exploits the fact that the video and audio signals are always directly correlated, unlike our case where partial observability is common. Finally, multiple prior works assume that synchronized audio-video signals are available and construct joint audio-visual feature representations for tasks such as source separation or sound classification [70, 71, 72]. While some of these works use artificial time shifts between the audio and video channels as a means of data augmentation, they have not been utilized for signal synchronization. Our work focuses on estimating the time shift between video and sensor streams.

2.3.2 Explicit Synchronization

A wide range of signal capture solutions have been designed which enforce synchronization at the hardware-software level. Here, we focus on three approaches. The first approach is used in sensor networks [73, 62, 74], including body area networks. Since all sensors are on the same network, protocols can be used to keep the sensor clocks synchronized, and corrections can be applied to address clock drift or skew [62]. In the second approach, all sensor signals can be wirelessly transmitted to a centralized collection node, such as a smartphone, where they are time-stamped to a common clock, thereby achieving synchronization (mCerebrum [75] is a representative example). These two approaches do not work for wearable cameras due to lack of network support, network bandwidth, and battery limitations. One exception is when all data collection takes place in the same location. In [76], a smartphone holder was installed in the location where tooth-brushing occurred, allowing video to be recorded on the smartphone camera (the centralized node) itself, thereby achieving synchronization. In [77], cooking activities were captured in the lab, enabling a wearable camera and other sensors to be synchronized via hardware (e.g., using genlock

where a reference signal from one device is used to synchronize all other devices). Note that we use the dataset from [77] for the experiments in Sec. 5.4.5. The third approach uses special hardware to achieve real-time synchronization [78, 79]. In [79], a periodically blinking LED is controlled to provide cues to synchronize different modalities. While such an approach can be effective, it requires additional implementation and system complexity, and the automated detection of the LED signal may be challenging in uncontrolled environments. Our approach leverages commodity hardware and standardized research-grade mHealth solutions to support a broad range of study designs.

2.3.3 Participant-Based Methods

The clapperboard approach to audio-visual synchronization provides a reliable solution because it introduces an explicit synchronization point which is visible across modalities. Analogous approaches exist for other multi-sensor synchronization tasks. In Plötz *et al*. [80], specific hand gestures were assigned to participants to provide explicit synchronization points for aligning video and accelerometer data. Similarly, Han *et al*. [81] propose a method to synchronize video and sensor data for walking behaviors by detecting and matching the maximum backward swings of the leg. Bannach *et al*. [82] develop a method to automatically detect specific gestures (e.g., 'clap') assigned to participants. These approaches can work in controlled settings, but they introduce additional participant burden and a single point of failure in the mobile setting.

2.3.4 Manual Synchronization

In cases where alternative synchronization approaches fail, a fall-back solution is to use tools such as ELAN [83] or Chronoviz [84] that enable the manual identification of synchronization points via inspection. This approach has been used routinely in prior mHealth and mobile sensing works and should be considered the default method [85, 86, 87, 88, 89]. Our goal is to remove the need for such manual efforts and provide a fully-automatic

solution to this important practical problem.

CHAPTER 3

FIRST-PERSON ACTION DECOMPOSITION AND ZERO-SHOT LEARNING

3.1 Introduction

First-person action recognition is the problem of recognizing an action using video which is captured from the first-person point of view, i.e. using a camera attached to the subject's head which can image the scene in front of them. The unique vantage point provided by a head-worn camera results in videos with two unique properties: 1) The camera viewpoint is implicitly guided by the attentional processes of the subject as they move through their environment to accomplish a task; 2) Activities involving hand-eye coordination, such as meal preparation, operating smartphones, and other activities of daily living, are captured in significant visual detail. In particular, one or both hands are often visible along with the object that is being manipulated to perform the action. As a consequence, the first-person vision paradigm can be a useful platform from which to study basic issues in activity recognition, such as the relationship between movement features (action models) and appearance features (object models) in recognizing an action such as "spread peanut butter."

In this chapter, we address the decomposition of first-person actions into a verb plus noun representation. Verb labels such as "take," "spread," and "open" are paired with one or more noun labels such as "cup," "peanut butter," and "coffee" to form an action model. Such a decomposition has been studied in [90], which primarily addressed action recognition from single images. There are two motivations for our work. First, from a practical point of view, it seems unlikely that the users of a first-person vision system are going to spend significant amounts of time training action models, even though action recognition is a potentially useful capability in supporting activities of daily living—for example, in elderly populations. It would therefore be valuable to predefine separate verb

and noun models for standard household actions and objects and then combine them with some tuning when adapting to a new environment. A zero shot learning approach is an appealing solution to this problem, where separately-trained noun and verb models could be combined with late fusion to recognize actions that have not been previously seen. We believe we are the first to address zero-shot learning through verb-noun decomposition in a first-person setting.

The second motivation for studying verb plus noun decompositions is to identify the roles and contributions of specific video features in modeling the verb and noun elements of an action. Many previous works have focused on detecting the objects which are utilized in actions as a means to perform recognition [21, 13, 91]. However, these approaches often exploit object context, and context often includes the hands that are holding or manipulating the objects. In these cases, the boundary between the noun and verb representations is unclear, particularly in the absence of pixel level segmentation. Likewise, in the case of verb modeling, our recent work [92] has demonstrated that classical motion features, such as dense trajectories, can be augmented with "first-person cues" such as the first person's head/hand movement, hand pose, and gaze information. These cues significantly improve the recognition accuracy. While motion features and first-person cues primarily encode the verb component of an action, the underlying visual features may well be exploiting appearance information from the objects being manipulated.

One can imagine two complementary strategies for achieving zero shot learning. The first, which is our approach, is based on the explicit construction of a decoupled representation for verb and noun components. Decoupling enables the use of late fusion to combine the models after they have been applied to an input video. Without explicit steps to construct a decorrelated representation, the learner could easily make mistakes which prevent generalization in the zero shot case. For example, when training a model for "take," its co-occurrence with "tomato" may mislead the model to learn to pick the features of red, round objects instead of the movement towards the subject which is characteristic of "take." Note

that when the visual feature representations are decoupled, the model can still leverage co-occurrence statistics for nouns and verbs which are derived from the knowledge of actions. For example, the fact that verbs like "open" and "close" co-occur with nouns like "door" and "window" but not "plate" can be used to prune the space of possible actions. A second strategy, exemplified by [33], is to leverage the co-occurrence statistics for nouns and verbs in a large text corpus to construct an embedding based on word2vec that relates the visual features for nouns and verbs. This approach could be effective when the embedding space accurately models the co-occurrence properties, but it requires sufficient training data to completely characterize the joint relationship.

Our approach of noun+verb decomposition provides a lens from which feature relationships can be observed and studied. By training separate verb and noun models it is easier to control the types of feature information that enter into the model. In this work, we use dense trajectories to characterize the verb concept and CNN features to characterize the object concept. By combining models using both early and late fusion, we can study the extent to which model tuning improves the performance, and analyze the extent to which visual features are shared between noun and verb models. Our overall approach is illustrated in Figure 3.1. Given a set of training video clips, our goal is to construct a model to infer the action label of a new video clip. We break this problem down into two steps: 1) Infer the noun and verb labels for an action as an intermediate task; 2) Combine the noun and verb labels to obtain action predictions. We show that this approach can take better advantage of the discriminative power of different features when predicting activity labels.

This chapter makes three contributions: 1) We propose a noun plus verb decomposition for egocentric actions which leverages the availability of egocentric cues to moderate the mapping between noun and verb models and specific low-level visual features; 2) We demonstrate promising experimental results for zero-shot learning of action models under both early and late fusion on standard egocentric action recognition datasets; and 3) As part of this work, we created additional annotations and ground truth labels for the GTEA

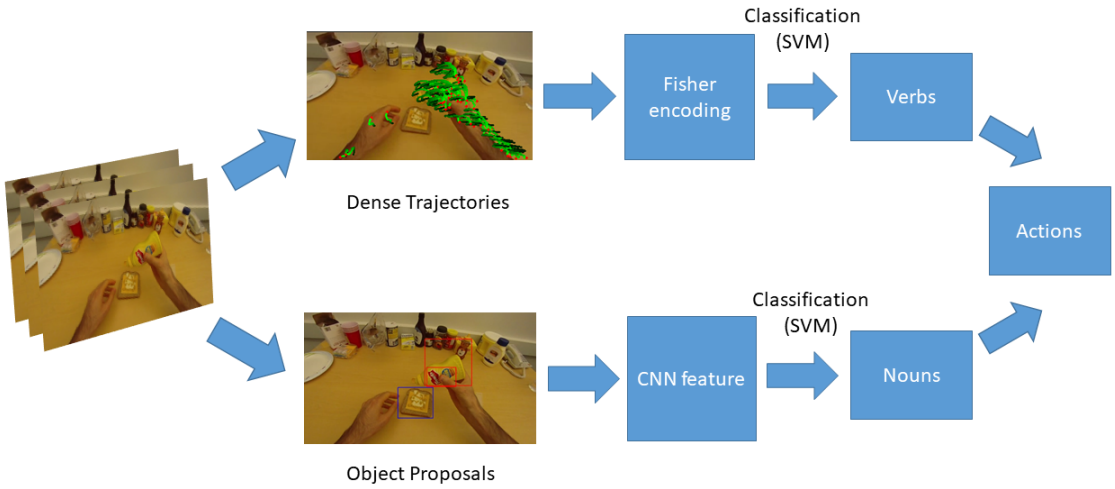


Figure 3.1: Overview of our model

datasets which are available in the project website.[1]

3.2 Formulation of Model Decomposition

The goal of our model is to classify the action label $a_i \in \mathbb{A}$ (*e.g.*, "pour water into cup", "spread peanut-butter on bread", "open coffee jar", etc) for a video clip i. An action label consists of:

- a verb $v_i \in \mathbb{V}$ (e.g. "pour", "spread", "open", etc), where $\mathbb{V}$ is the space of all discrete verbs.
- a sequence of nouns representing object names $n_i \in \mathbb{N}$ (*e.g.*, {"water", "cup"}, {"peanut-butter", "bread"}, "coffee", etc), where $\mathbb{N}$ is the space of all objects.

The verb v_i and noun n_i can be thought of as two attributes of an action a_i. We further denote the feature vector of video clip i as $\mathbf{x_i} \in \mathbb{X}^d$. Our goal is to learn a mapping function $f : \mathbb{X}^d \rightarrow \mathbb{A}$.

[1] `http://cbi.gatech.edu/fpv`

The label space $\mathbb{A}$ is naturally factorized into $\mathbb{V}$ (the space of all verbs) and $\mathbb{N}$ (the space of all nouns). We propose to utilize this structure to model first-person actions. Our key observation is that $\mathbb{V}$ is primarily encoded by motion features $\mathbf{x_v} \in \mathbb{X}^{d_v}$ across frames, while $\mathbb{N}$ is largely captured by object features $\mathbf{x_n} \in \mathbb{X}^{d_n}$ within frames. More formally, by decomposing the label space $\mathbb{A}$ into $\mathbb{V}$ and $\mathbb{N}$ and taking different features $\mathbf{x_v} \in \mathbb{X}^{d_v}$ and $\mathbf{x_n} \in \mathbb{X}^{d_n}$ for verbs and nouns, the classifier f_A is decomposed into two mappings $f_V : \mathbb{X}^{d_v} \to \mathbb{V}$ and $f_N : \mathbb{X}^{d_n} \to \mathbb{N}$ through a composite model $s : \mathbb{V} \times \mathbb{N} \to \mathbb{A}$:

$$f_A(\mathbf{x_v}, \mathbf{x_n}) = s(f_V(\mathbf{x_v}), f_N(\mathbf{x_n})), \tag{3.1}$$

where f_V and f_N map motion features and object features into verbs and nouns, respectively, and s maps the combination of a verb and a noun to an action.

This factorization scheme has two major advantages. First, it enables zero shot learning. We can recognize a novel action class where the attributes (verb or noun) occur in the training set, but the combined action label pairs do not. Second, it allows us to explore how motion and object features contribute to first-person action recognition. We now describe our motion and object features and our inference model.

3.2.1 Features

We use improved dense trajectory features to model motion and CNN features to model appearance. We denote them as $\mathbf{x^{idt}}$ and $\mathbf{x^{cnn}}$. These two features provide complementary information in predicting action labels. We will compare their discriminative power under different models in Section 3.3.

Motion Features

To build feature $\mathbf{x^{idt}}$, we implement a variant of the improved dense trajectories (IDT) feature proposed by Wang and Schmid [93]. We add the positions of the tracked points to the local descriptors as in [94]. IDT combines explicit camera motion compensation, dense tracking of local interest points and the stacking of multiple descriptors along the tracked

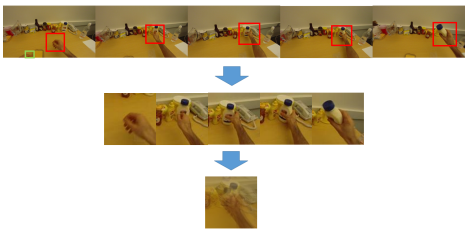


Figure 3.2: Object proposals around manipulation points

Figure 3.3: Additional object proposals found near hands

trajectories. We use the same feature set as in [93], including trajectory features, HoF, HoG and MBH to capture shape, appearance and motion information and encode them into Fisher vectors. Motion boundary histograms (MBH) rely on differential optical flow, and are more robust to camera motion than optical flow. MBH together with camera motion compensation make IDT a valuable feature for first-person action recognition. We encode the local descriptors in each channel (HOG, HOF, etc.) into a fisher vector and concatenate the vectors from all channels as in [95]. We reduce the dimensionality of the descriptors by half and choose 10 clusters for optical flow trajectory feature and 40 clusters for the rest of the features for fisher encoding.

Appearance Features

To build features specialized for classifying objects, we use the activations from a convolutional neural network. This feature has shown excellent performance on object classification and detection in images [96, 97, 98, 99] and videos [100, 94, 101]. Unlike presentations of whole frames in previous works, we extract features $\mathbf{x}^{\mathbf{cnn}}$ guided by hand positions.

We divide the hands into three types: right hands, left hands and intersecting hands. The possible combinations of the hand types are constrained. At most two types of hands are possibly present in a frame. Also intersecting hands cannot happen at the same time as right hands or left hands. Object features are built for each type of hands.

In order to find the hand position and classify the hand types, we need to segment the hands. Textonboost, implemented in [102], is used to find the hand and arm regions in an image. Other hand segmentation and tracking techniques can also be used to this end. Area, centroid, orientation, major axis length, minor axis length, and extrema are computed for each connected component in the resulting mask as regional geometric features, which are used to classify the hand types. Through the application of an RBF-kernel SVM [103] to the regional geometric features of each connected component, a model is trained to label each connected component as right hand, left hand or intersecting hand. Hand position is found for the three types of hands, and the manipulation point is identified as in [104].

We utilize the hand information to build noun representations in two ways: First, we select a proposal around each manipulation point in each frame and pool the CNN features across the video clip. We concatenate features for each hand type (left/right/intersecting hand) to form a feature vector. Figure 3.2 illustrates how the proposals around the manipulation points are pooled across a video clip. In addition, we encode the CNN features of the proposals in the vicinity of hands into a Fisher vector.

To build the Fisher vector, we extract object proposals using edge boxes [105]. Figure 3.3 shows an example frame with proposals found near hands. We use R-CNN [97] to extract CNN features from the resulting proposals using the seventh layer of AlexNet pretrained on ImageNet ILSVRC 2012. These 4096 dimensional feature vectors are reduced in dimensionality by PCA to 41 dimensional vectors and encoded into a Fisher vector using 50 clusters.

3.2.2 Inference Model

We evaluate the performance of our models for conventional action recognition and zero-shot action recognition. The models described in this section can be utilized with different types of features. The details are provided in Sec. 3.2.2.

Action Recognition

We build direct and factorized models to classify the actions. In a direct model, an action classifier $f_A(\mathbf{x_v}, \mathbf{x_n})$, which is a mapping from the motion and appearance feature spaces to the action label space, is learned. In a factorized model, a separate verb classifier $f_V(\mathbf{x_v})$ and a noun classifier $f_N(\mathbf{x_n})$ are learned. They are then combined via the mapping function s. The details of a factorized model are described as follows.

In this work, linear SVMs [106] are used to classify verbs and nouns. To combine the results of the two models to obtain the action label a_i, we assume that the nouns and the verbs are independent. The action model can then be written as

$$P(a_i|x_{vi}, x_{ni}) \approx \sum_{v_i=1}^{V} \sum_{n_i=1}^{N} P(a_i|v_i, n_i) P(v_i|x_{vi}) P(n_i|x_{ni}), \tag{3.2}$$

where x_{ni} and x_{vi} are the noun and verb feature vectors for video *i*. We model $P(v_i|x_{vi})$ and $P(n_i|x_{ni})$ using linear SVMs. $P(a_i|v_i, n_i)$ defines the action label space for the data set. During testing, the action label a_i for video v_i is chosen to be the label that maximizes $P(a_i|x_{ni}, x_{vi})$.

In Equation 3.2, we made two assumptions: 1) $P(v_i|x_{vi}, x_{ni})$ and $P(n_i|x_{vi}, x_{ni})$ are independent; 2) v_i and x_{ni}, n_i and x_{vi} are independent. The first assumption corresponds to the independence of verbs and nouns. The second assumption relies on verb/noun feature decorrelation. These two assumptions are the key to our factorized approach to zero-shot learning. In the experiments, we will show that properly choosing features that approximately satisfy these two assumptions is critical to both conventional and zero-shot learning.

Fusion of Motion and Object Information

We investigate several ways to combine the information from IDT and CNN features $\mathbf{x^{idt}}$ and $\mathbf{x^{cnn}}$. Figure 3.4 shows the three approaches we use in this work – combining them before classifying (early fusion, dotted arrows), combining the results from classifying the attributes (late fusion, solid arrows), and performing both (combined early and late fusion,

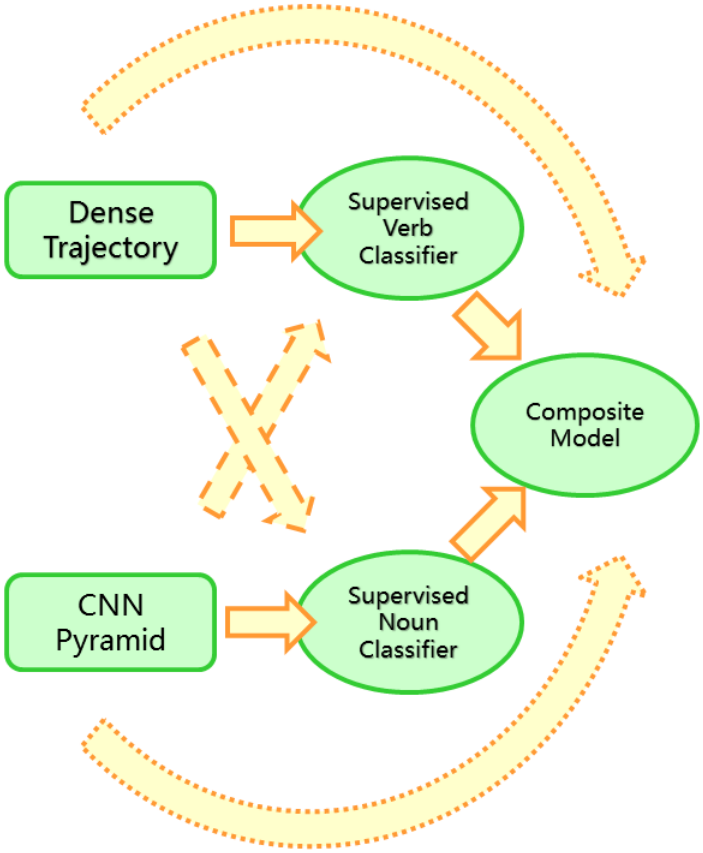


Figure 3.4: Different information fusion schemes

solid arrows together with dash arrows). Note that related early fusion and late fusion schemes have been evaluated for multimedia semantic indexing [107, 108].

Early Fusion: We combine the two features by concatenating them and passing the combined feature to a linear SVM to classify the action labels.

$$\mathbf{x} = [\mathbf{x^{idt}}, \mathbf{x^{cnn}}] \tag{3.3}$$

Late Fusion: We use $\mathbf{x_v} = \mathbf{x^{idt}}$ and $\mathbf{x_n} = \mathbf{x^{cnn}}$ to fit linear SVM models $f_V(\mathbf{x_v})$ and $f_N(\mathbf{x_n})$. We then fuse the two models through the inference model in Equation 3.2.

Early Fusion + Late Fusion: We take the feature $\mathbf{x_v} = \mathbf{x_n} = \mathbf{x}$ as in early fusion, and train verb classifier $f_V(\mathbf{x})$ and noun classifier $f_N(\mathbf{x})$ with the joint feature. We then fuse the two models using the inference model in Equation 3.2.

Early fusion brings information from both features into the machine learning model at the feature construction stage. It implicitly encodes the correlation between verbs and nouns. This correlation could be helpful in conventional action recognition but is not desired in zero-shot learning, as we will show in Section 3.3. Late fusion and early + late fusion are compared for zero-shot learning.

3.3 Experiments

We evaluate our model under two experimental conditions: conventional action recognition in Section 3.3.2 and zero-shot learning in Section 3.3.3.

3.3.1 Datasets

We evaluate the performance of our method on two first-person action datasets – GTEA and GTEA gaze+.[2] In the GTEA dataset, 525 video clips on 71 actions are cropped from 28 videos of meal preparation on seven recipes performed by four subjects. There are 10 classes of verbs and 38 classes of nouns. We performed cross-validation in a leave-one-subject-out fashion. In the GETA gaze+ dataset, six subjects are asked to perform a set

[2] `http://cbi.gatech.edu/fpv`

of seven meal preparation activities, resulting in 37 videos. Each video contains approximately 100 activities. We provided additional annotations for the verbs in this dataset so we can use motion features to distinguish verbs with the same name but different semantics. For example, the action "open" provides different motion patterns in the action "open a bottle" and "open a fridge." After the rare (number of instances less than 3) and short (number of frames less than 6) activities are eliminated, 1947 clips on 44 activities are retained for training and testing. There are 15 classes of verbs and 27 classes of noun sequences. This dataset contains a larger variation in backgrounds and more diverse object appearances. We did a leave-one-subject-out cross validation on those subjects who performed all seven activities.

3.3.2 Conventional Action Recognition

Table 3.1: Average accuracy over classes for conventional action recognition.

		GTEA			**GTEA Gaze+**		
		Verb	**Noun**	**Action**	**Verb**	**Noun**	**Action**
1	**IDT**	0.7555	0.4507	0.4041	0.6674	0.5345	0.5126
2	**CNN**	0.5954	0.6494	0.5308	0.5865	0.6020	0.4489
3	**Early Fusion**	0.7939	0.6001	0.5567	0.7507	0.6562	0.5779
4	**Late Fusion**	-	-	**0.7254**	-	-	0.5495
5	**Early+Late Fusion**	-	-	0.6059	-	-	**0.6026**
6	**Li et al. [92]**	-	-	0.621	-	-	0.605

The state-of-the-art results on our two datasets were achieved by [18] by the time this work was published. However, [18] used additional annotations for the locations of the objects of interest. Therefore, we compare our results to our previous performance in [92]. We configure our experiments following the experimental settings from [92]. On GTEA, we perform leave-one-subject-out cross-validation on the four subjects. On GTEA Gaze+, we perform leave-one-subject-out cross-validation on the four out of six subjects who completed all seven recipes. In GTEA dataset, the position of the subject is mostly

fixed, whereas GTEA Gaze+ involves a larger motion of the subject and thus a more diverse background, illumination conditions and object appearance. The hands also appear more often and are detected with higher accuracy in the GTEA dataset.

Table 3.1 shows the accuracies for verb labels, noun labels and action labels when using different models. Reported accuracies are obtained by training and testing SVM classifiers using the features and fusion methods described in the table and averaging the testing accuracies across all verb, noun and action classes. The first two rows of the table show the results of using either IDT or CNN features alone to make predictions. The next three rows (3 to 5) give the performance for different fusion methods. The first observation is that, not surprisingly, the action classification performance in rows 1 and 2 is lower than that for rows 3-5, demonstrating that the combination of verb and noun information is important for action prediction. We can gain insight into the correlation of IDT and CNN features with the verb and noun classes by comparing the results for verb and noun prediction across rows 1 and 2. We find that CNN features are better at classifying nouns (0.65 vs 0.45 in GTEA, and 0.60 vs 0.53 in GTEA Gaze+), and IDT features perform better at classifying verbs (0.76 vs 0.60 in GTEA, and 0.67 vs 0.59 in GTEA Gaze+). The relatively poor performance of IDT features on noun classification and CNN features on verb classification supports the second assumption in Sec. 3.2.2, that v_i and x_{ni}, n_i and x_{vi} are independent.

We note that the performance gap between the two features for both noun and verb classification is larger in GTEA than in GTEA Gaze+. This difference in performance can be explained by the characteristics of the two datasets. Note that the CNN classifier works equally well for verbs in both datasets. This is reasonable because object information used by CNN bounding boxes will be equally discriminative across the two conditions. When CNN features are used to predict verbs and nouns, we use the same CNN bounding boxes. We observe a significant performance drop for IDT on GTEA Gaze+. This is because GTEA is collected under a more controlled experimental setting. The subject is still for

most of the time, and the actions are performed in similar ways. In contrast, GTEA Gaze+ is collected in a more natural setting. The subjects move around during the cooking activity and perform the same action in various ways, making verb prediction more challenging.

Row 3 shows the performance of early fusion, which combines IDT and CNN features in predicting the class labels for each column. Examining the noun and verb classification results, we can observe that it is generally beneficial to combine IDT and CNN features when learning both verb and noun classifiers, even though they are specialized for different label types. In both GTEA and GTEA Gaze+, verb classifiers incorporating CNN features are more accurate than those trained with only IDT features. This discrepancy makes sense as the training set will capture the co-occurrence of nouns and verbs, which can provide additional information. Likewise, in GTEA Gaze+, when the noun classifier is trained with the early fusion features, it performs better than the one trained with CNN features alone. This result likely occurs because in GTEA Gaze+, fewer hands are detected. Moreover, in approximately 10% of the instances, very few proposals were detected (no more than 5 edge boxes per video near the hands). In those cases, guiding the CNN features by hand position is not sufficient to classify the nouns, and the appearance information in dense trajectories can provide complementary information for classifying the objects. As we will see later in the zero-shot learning experiment, utilizing this correlation by modeling verbs and nouns together will harm the model's ability to generalize to novel actions. In this case, using IDT features for verbs and CNN features for nouns can better decompose the feature space and achieve better results.

We now compare the performance of action classification across rows 3-5. Note that the late fusion methods (rows 4 and 5) consistently outperform early fusion. Since this is not a zero-shot learning experiment, we might expect the best performance from early fusion, since it has the most flexibility in modeling the co-occurrence of noun and verb information. We believe that the underperformance of early fusion is due to the limited number of training examples for each action. Early fusion builds a separate classifier for

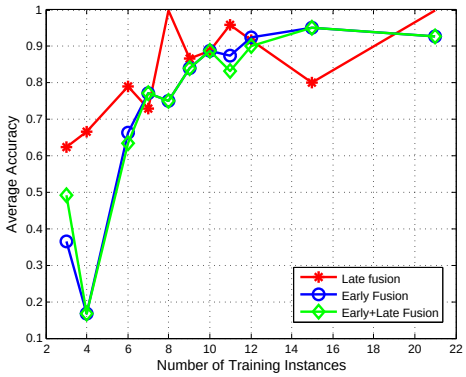


Figure 3.5: Accuracy vs number of training samples per class for GTEA

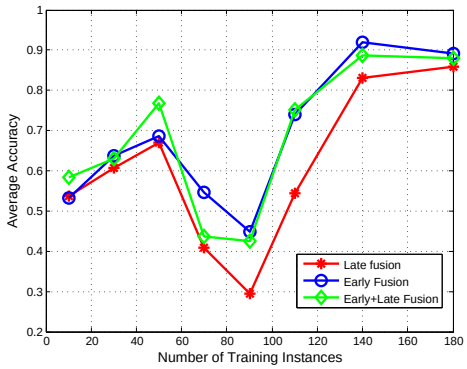


Figure 3.6: Accuracy vs number of training samples per class for GTEA Gaze+.

each noun+verb pair, while late fusion can utilize noun and verb models in which examples are pooled across action categories, leading to a larger training set. Notice that for GTEA Gaze+, early+late fusion gives the best result, while late fusion is most accurate for GTEA. This result can also be explained, in part, by differences in dataset size. Early+late fusion combines IDT and CNN features when predicting noun and verb labels prior to late fusion. When there are a small number of training examples, it is difficult to capture the co-occurrence pattern and the resulting classifier is less accurate. This difficulty is illustrated in Figure 3.5, which shows the accuracy of predicting the action label in GTEA as a function of the number of training samples per class. The accuracy is averaged over all of the classes that have the same number of training instances and across all splits of cross-validation. Note that late fusion dominates early+late fusion and early fusion for small numbers of samples. Figure 3.6 plots the same variation for the GTEA Gaze+ dataset. Note that the number of training samples is substantially larger. In this case, the performance of early fusion and early+late fusion is comparable, and both are superior to late fusion overall. On average, GTEA has 5.55 training instances per action, whereas GTEA Gaze+ has 35.32, which explains why GTEA benefits more from late fusion.

Row 6 compares the result to our previous result in [92]. We outperform our prior method in the GTEA dataset by approximately 10% and achieve comparable performance

in GTEA Gaze+. Note that the method in [92] includes LAB color and LBP descriptors to capture appearance and texture information in the dense trajectory feature. This approach will help in conventional action recognition but will also make the model more susceptible to confounding effects (correlation between verb and noun labels) when classifying verbs in zero-shot learning.

3.3.3 Zero-Shot Learning

We conducted a zero-shot learning experiment to assess the ability of our model to disentangle the effects of verbs and nouns. We chose those actions whose verb and noun labels were present in multiple actions, resulting in 45 test actions in GTEA and 26 actions in GTEA Gaze+. Among these actions, the verbs in GTEA for testing are *open*, *close*, *take*, *put*. The verbs in GTEA Gaze+ are *open* (door, etc.), *close* (door, etc.), *take*, *put*, *open* (container, etc.). In GTEA and GTEA Gaze+, 13 and 12 nouns are used for testing, respectively. In each round, we hold out one action for testing and use the remaining actions for training. The accuracy is averaged over all runs. Note that the training set includes all of the actions in the original dataset except the held-out action. The testing space for the model comprises all the actions, including the held out action. This approach is in contrast to many previous zero-shot learning models which only differentiate between unseen classes [29, 24, 25] in the held-out dataset. We restrict the samples for testing to the novel actions because the goal of this section is to evaluate the ability of this model to classify novel actions. If we test samples from the whole action space, the results will be dominated by previously seen classes, which are evaluated in Sec. 3.3.2. The average accuracy over all held-out actions for two factorized models is reported for both datasets.

Table 3.2 shows the results for zero-shot learning. Each row is the result of a leave-one-action-out experiment. Note that the verb and noun columns are not results for individual classifiers. They show the accuracies of verbs and nouns in the final late fused action predictions. This table shows that late fusion performs significantly better than early+late

Table 3.2: Average accuracy over classes for zero-shot learning. Chance-level action recognition accuracies for the two datasets are 0.014 and 0.0385, respectively.

	GTEA			GTEA Gaze+		
	Verb	**Noun**	**Action**	**Verb**	**Noun**	**Action**
Late Fusion	0.8917	0.6581	**0.6870**	0.3248	0.5340	**0.2919**
Early+Late Fusion	0.8528	0.4065	0.3963	0.2768	0.4344	0.1598

fusion. This result supports the importance of decorrelated verb and noun models. The co-occurrence information between verbs and nouns captured by the early fusion feature may help in a conventional action recognition task, as shown in the results for GTEA Gaze+ in Table 3.1, where both the IDT and CNN features are discriminative for both labels. However, it will harm the result in zero-shot learning, since we need the model to be able to find the correct verb and noun labels independently, in order to predict the novel verb-noun combination in the test set. Therefore, choosing features that satisfy the assumptions in Section 3.2.2 is important. In future work, we plan to investigate automated methods for constructing decoupled models for verbs and nouns in a complex dataset such as GTEA Gaze+.

The low performance in GTEA Gaze+ could stem from the correlation between verbs and nouns, which can be verified by examining the predicted labels. The verb labels are frequently predicted as the opposite action ("put" is predicted to be "take" and vice versa). Since they usually are paired with the same objects, the features learned by the "take" classifier might include object information, thereby leading to the misclassification of the novel action.

3.4 Conclusion

We developed a verb+noun model for zero-shot action recognition in first-person videos. We argue that the coupling between visual representations of nouns and verbs, which leads to good performance in conventional action recognition, is detrimental to zero-shot learn-

ing. This is due to the need for composable noun and verb models that support late fusion. Late fusion makes it possible to combine previously trained noun and verb models to describe novel actions. For example, if a verb model incorporates visual features from nouns seen during training, then it will have a more difficult time generalizing when used in an action model containing novel nouns. A similar argument applies to noun models. We investigated the extent to which existing visual feature representations for noun cues (CNN features) and verb cues (improved dense trajectory, IDT, features) could be used to construct a decorrelated representation. Our results demonstrate that while CNN and IDT features are not truly decorrelated representations of noun and verb information, they can be used successfully to develop a late-fusion approach to zero-shot learning. We introduced a variant of late-fusion called early+late, which pools visual features in training separate noun and verb models. This approach makes it possible to exploit the co-occurence between noun and verb cues when building separate concept classifiers. It leads to better performance in conventional action recognition, when sufficient training samples exist and the training set covers the same concept class as the testing set. However, its poor performance in the zero-shot case demonstrates the importance of constructing decoupled representations when utilizing late fusion (see Table 3.2). We believe that the egocentric domain is a particularly useful vantage point from which to study zero-shot action recognition, as well as the more general question of how noun and verb models differ and what they should ideally describe. For example, the fact that the hands are often visible at sufficient resolution to support segmentation and motion analysis makes it relatively straightforward to capture verb cues derived from the hands and to study noun-verb interactions in the context of object manipulation.

CHAPTER 4
SCREEN-USE ACTIVITY DETECTION AND ASSOCIATED ATTENTION ESTIMATION

4.1 Introduction

TV watching is an important part of many people's lives, but excessive screen time is linked to patterns of sedentary behavior and can be associated with negative health outcomes. The American Time Use Survey reported that in 2016, Americans spent an average of 2.73 hours per day watching TV. This was the third most time-consuming activity following sleeping and working [109]. Research has demonstrated a link between TV watching, sedentary behavior patterns, and obesity in adults and children [110]. TV watching is also negatively correlated with children's verbal abilities and other aspects of physical, cognitive, and emotional development [111, 112, 113, 114, 115, 116]. Takeuchi *et al.* investigated the effect of TV watching on the development of brain structure in children. They found that the Verbal Intelligence Quotient (VIQ) was negatively effected by TV watching [117].

TV watching is also linked to unhealthy eating and alcohol use behaviors. Northup [118] and Harris and Bargh [119] established a link between TV watching and unhealthy eating. Mejia *et al.* and Morgenstern *et al.* found that exposure to alcohol advertising content or alcohol use in films is associated with alcohol use and binge drinking in adolescents [120, 121]. Current research connecting exposure to TV content to health outcomes is based almost entirely on self-report data obtained by administering questionnaires. While this is sufficient to establish qualitative associations between TV exposure and health risk, these methods are unable to map the detailed structure of TV exposure in terms of when and for how long participants attend to screens in their environment. As a result, they cannot

determine which patterns of screen viewing and which sources of media content present the greatest sources of risk.

In this work, we describe a ubiquitous sensing system which can automatically map the fine-grained structure of a participant's attention to multiple screens in their environment over time. Our approach is enabled by the emergence of low-cost, wearable, head-mounted camera systems which can capture Point-of-View (POV) video. These POV cameras make it feasible to capture a continuous record of a person's visual inputs (visual exposure) as they go about their daily life. We develop a machine learning-based video analysis system which can automatically detect the screens in a participant's field of view and identify when one or more screens are being watched. Given a POV video file as input, our system outputs a complete record of the onset and offset of all screen-watching episodes. Our analysis software and attention models are freely available to the research community and constitute a novel tool for studies involving screen-watching behavior. In addition, our promising results for detecting and tracking attention to screens is a step towards the development of a mobile health intervention, which could be triggered when unhealthy patterns of TV watching are detected.

In contrast to our wearable camera approach to monitoring TV watching, past research has explored alternative strategies to detect screen-watching behavior using different sensors and/or by instrumenting TVs instead of the TV viewers. An early example is the Unitam meter [122], a commercial system that estimates audience sizes for TV programs. It consists of a data-collection device connected to a TV along with a special remote control. Participants can register their presence as TV watchers using the remote control, and the data collector monitors which content is being displayed. However, systems that require users to explicitly register can suffer from under- and over-reporting biases. On the other hand, approaches based on color sensors [43] and proximity sensors, such as RFID tags [123], eliminate the need for the users to register manually but suffer from the problem that TVs may be on even when no one is attending to them. A potentially more accurate

solution is to instrument screens with outward-facing cameras so that screen-watching behaviors can be detected automatically from the TV's "point of view" [35]. In the case of displays which are fixed in the environment, such as conventional television sets, this approach has the advantage that the camera sensor can be continuously powered, thereby removing concerns of battery life. However, the screen camera approach suffers from three limitations. First, there is an important social element to TV watching, with families and friends frequently watching a single screen together. Keeping an accurate account of individual TV exposure in these scenarios would require a complex and potentially unreliable process of tracking the identities of individual TV viewers over time. Second, the quantity and variety of screens is proliferating, with users accessing content from a variety of mobile devices as well as computers and televisions. Maintaining an accurate and complete record of exposure would require the utilization of cameras on all of these devices and the coordination and sharing of TV exposure records across devices, a potentially challenging undertaking. Moreover, bars and clubs can be a significant source of exposure to TV content with associated health risks, and they are unlikely to incorporate a TV exposure measurement system. Finally, privacy concerns are paramount in any situation in which personal records of media consumption are involved. With screen-based cameras, participants would require a high level of trust to believe that they are in control of what is being recorded at all times. With a solution based on wearable cameras, in contrast, participants can ensure that they are not being recorded by simply taking off or turning off the camera.

Our TV monitoring system consists of a wearable video recorder which captures POV video and an analysis pipeline consisting of a TV detector and an attention classifier. Our TV detector is based on the state-of-the-art Faster RCNN method. It is capable of detecting screens of various types with a mean average precision (mAP) of 80% on the TV/monitor class of the PASCAL VOC 2007 test dataset (a standard benchmark for object detection in computer vision). Our attention classifier identifies attention to screens with a precision of 91.69% and a recall of 94.51% on a TV-watching dataset collected in a home environ-

ment in Section 4.7.4. In another experiment, discussed in Section 4.7.5, in which subjects switched their attention between multiple devices, the attention classifier achieved a precision of 98.13% and a recall of 87.07%. We also tested our system in a natural environment in Section 4.7.6, in which our system achieved a precision of 87.14% and a recall of 81.68%. As a part of this work, we will release a new dataset that consists of 60 video clips with annotations of TVs and tablets and another dataset consisting of 13 hours of first-person TV-watching videos with attention annotation. We believe that we are the first to tackle the TV-watching problem using a wearable camera system. Note that we use the term "TV" to denote any object which has a screen for viewing, including conventional televisions, desktop PC screens, laptops, tablets, and smartphones. This work makes the following three contributions:

- We develop and validate a completely automated system for detecting the onset and offset of bouts of TV watching within daily life activities. Our approach does not require any annotation by the participant and can map the temporal structure with which attention to screens is intermixed with social interactions, eating, and other activities.

- We demonstrate that attention to screens can be estimated using only a wearable video recorder, and in particular does not require eye-tracking technology. We develop a computational model of attention during TV watching and demonstrate the ability to accommodate attention to different types of screens, such as televisions and tablets.

- We provide three new datasets to the research community.[1] The first is a dataset of screens "in the wild" to support the development of detection algorithms. It consists of video clips and images with manual annotations of TVs and tablets. The second dataset consists of POV videos collected in the Aware Home laboratory at Georgia

[1]The datasets and software for this project are available from the project website: `https://yunzhang07.github.io/tvwatch`.

Tech, with frame-by-frame annotations of TV-watching behaviors. The third dataset comes from a protocol in which subjects shift their attention between a TV screen and a handheld tablet. This dataset also comes with frame-by-frame annotations.

4.2 Approach

Our approach to detecting TV-watching behavior using only a single wearable camera, *without an eye tracker*, leverages our understanding of visual attention from the psychology literature [60, 59, 58, 57] and our prior work on gaze prediction [61]. These works demonstrate that when people are engaged in specific tasks, their patterns of attention can be predicted from the structure of the task environment and the objects and visual stimuli that are most relevant to the task. In our context, this background leads us to make three assumptions that form the basis for our approach:

- During extended periods of TV viewing, subjects will remain fixated on the TV screen for significant periods of time, which we refer to as *bouts*.
- In-between bouts of TV viewing, subjects may direct their gaze towards other objects, such as food items, social partners, and phones.
- Bouts of TV viewing may also be punctuated with changes in position, such as a posture change, that bring the TV screen into a new location in subjects' field of view.

The empirical studies we have conducted, as well as the success of our approach, provide additional empirical evidence for the accuracy of these assumptions.

At the heart of our approach is a simple observation: When subjects watch TV for extended periods of time, the TV occupies a relatively stationary position within the subject's field of view, and their head movements are minimal. See Figure 4.2 for a demonstration of this finding in our dataset. Thus, to a first approximation, if we can detect the presence of a TV in the field of view over a contiguous period of time in which the subject's head

is still, then we can assume that the TV is likely to be the attentional target. In this regard we are helped by an aspect of our wearable camera platform which is normally viewed as a deficit: Namely, the inherently limited field of view of the Pivothead video recorder, which is only 77 degrees in the horizontal and 43 degrees in the vertical. Since screens are ubiquitous in many environments, a wearable camera with a large field of view would be likely to incidentally capture screens during many other activities of daily living, such as reading, eating, or knitting. However, due to the design of the Pivothead, it is unlikely that a screen will occupy a significant portion of the field of view for an extended period of time while *not* being the subject of attention.

However, simply counting the number of frames in which TV screens are detected within the field of view is insufficient for three reasons. First, it can be difficult to distinguish posture changes, which shift the TV to a different position within the first-person video without changing the TV-viewing behavior, from shifts to other objects such as food, which do represent a departure from TV viewing. Second, it is relatively common to watch TV while holding a phone or a tablet, and even with the limited field-of-view of the Pivothead, the secondary screen can appear sporadically within the video. Third, our long-term goal is not merely to determine if TVs are being watched but also to determine what types of content are being attended to. For these reasons, it is advantageous to explicitly model the focus of attention within the camera field of view. This approach allows us not only to determine that a TV screen is being watched but to identify to which screen we are referring and to select among multiple screens that may appear simultaneously. We refer to our estimate of where the subject is currently attending as the *center prior*, mirroring the use of that term in the literature where it refers to an average fixation point integrated over many frames of data [61]. Without an eye tracker, we have no hope to predict the subject's precise locus of attention on a frame-by-frame basis. But under the assumption of bouts of TV viewing, the center prior becomes a useful proxy for the subject's attention.

Our approach, then, consists of two analysis loops. First, there is an *adaptation loop*,

which operates at the time scale we associate with bouts of continuous TV viewing. Based on our data and the literature, we establish 30 seconds as the bout duration. Over the bout duration, we continuously re-estimate the position of the center prior by integrating the detected positions of TV screens that were present in approximately the same position. Intuitively, TVs that remain in approximately the same position while the head is stationary will capture the center prior over time. The second, inner loop, is the *detection loop*, which operates on a six-frame window of video and detects TV-watching behavior at the window level based on multiple cues: head motion, the position of detected TV screens, and the current estimate of the center prior. We train a classifier for the inner TV-watching detection loop and use a conditional random field (CRF) architecture to smooth the data. The block diagram for the inner detection loop is given in Figure 4.1. Detected TV screens and the center prior combined with head motion estimates form the input feature vector for our detector for TV watching.

Our approach relies on the ability to reliably detect TV screens in natural environments within the home and office. Note that we use a broad definition of TV screen that includes tablets and monitor screens as well as more traditional television sets. Our approach to TV detection uses the Faster RCNN object detection system [124], which has been shown to deliver excellent object detection performance at a reasonable computational cost. We fine-tuned the network for recognizing TVs using the data available in public object detection datasets along with a new TV screen video dataset that we collected. The original system, consisting of a generic region proposal network and an object detection network, becomes a two-level cascade object-specific network. We then use a Kalman filter to smooth the TV detection results.

We now describe each of these components of our solution and its evaluation in more detail. In Section 4.3, we describe our method for reliably detecting TV screens and performing temporal smoothing. In Section 4.4, we describe our inner loop attention classifier, which identifies TV-watching behavior. In Section 4.5, we present our outer loop attention

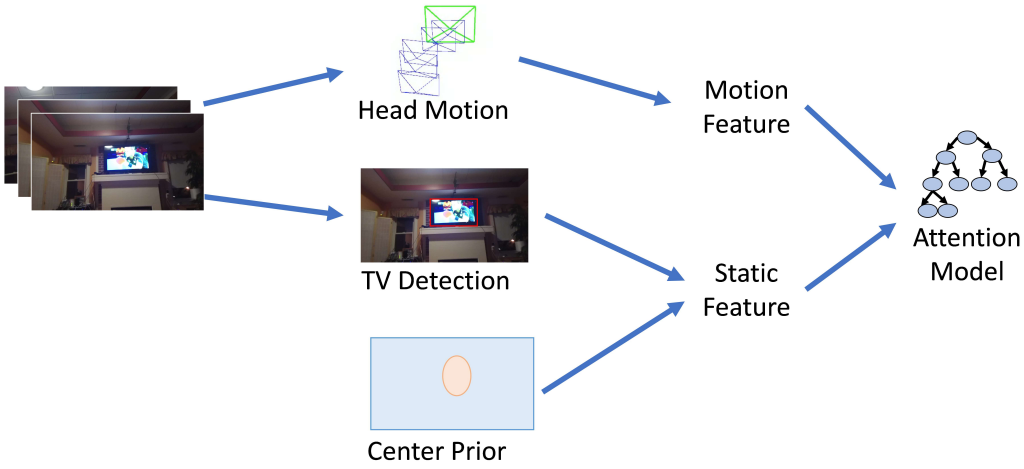


Figure 4.1: Overview of our detection system, which operates as an inner loop on a six-frame window of video and classifies TV-watching behavior using a combination of static and dynamic features. Note that the center prior models the subject's attention and is modifed through an outer adaptation loop.

model for adapting the center prior. The implementation of these components in a reference system for detecting TV watching is described in Section 4.6. An experimental evaluation of our system is presented in Section 4.7.

4.3 TV Detection

The starting point for our work is the reliable detection and tracking of TV screens in each frame of the video. This analysis allows us to segment the input video into sections in which a TV screen is visible and can be tracked continuously from frame to frame. We would expect to obtain such contiguous sections of TV screens during bouts of TV viewing. We begin by detecting TV screens in each frame using a state-of-the-art object detector. This step is followed by a temporal smoothing step in which we track the detections across consecutive frames. Smoothing improves the detection results by removing false detections and filling in missing detections identified via temporal consistency. While these technologies have been used in many prior computer vision applications, we believe we are the first to develop and test such a TV screen tracking system using modern classification methods.

4.3.1 Single Frame Detection

The key step is to identify all of the TV screens within each first-person video frame produced by the Pivothead glasses. We use a state-of-the-art object detection system, Faster RCNN [124], to detect TV screens. Faster RCNN has demonstrated excellent performance on the PASCAL VOC [125] and COCO [126] datasets, which are standard datasets used in computer vision for training and evaluating object detection algorithms. Faster RCNN utilizes a deep convolutional neural network designed for multiclass object detection. It consists of two sub-networks: a generic region proposal network that generates candidate regions that are likely to be an object, and an object detection network to classify the candidate regions and predict the bounding box positions.

The use of a deep network to detect objects is critical because it can exploit the large amounts of training data that are available for common objects such as TV screens. The ability to perform feature learning with a large dataset has been shown to dramatically outperform prior methods based on hand-crafted features. While Faster RCNN can be trained to detect dozens of different object classes, for our application we are only concerned with detecting TV screens and rejecting any nonscreen objects in the user's environment. We address this by modifying the Faster RCNN method and retraining it for our purpose.

We adapt Faster RCNN by starting with the pretrained general purpose object detection network, which can be downloaded for the Caffee deep learning environment. We then restructure the output layer of the network and finetune it on an additional corpus of TV images. The network functions as a two-level cascade object detector. The region proposal subnetwork, which classifies the regions as foreground and background, now works as a first-level classifier for TV detection, since in this work the only foreground object class is TV. The trained TV detector produced better results on images with TV screens than the general purpose detector.

4.3.2 Temporal Smoothing

Given a sequence of frames in which TV screen detection has been performed, the next step is temporal filtering in order to smooth the detection results. We build a filter based on a simple online and realtime tracking algorithm (SORT) [127] for 2D tracking of multiple objects in video. SORT is an implementation of a Kalman filter, which performs temporal smoothing of the bounding box coordinates of the detected TV screens, and a data association filter, which connects contiguous detections into an object track and handles both missing detections and false positives. We have extensively modified the freely available SORT codebase to utilize state space and transition model that are tailored to our application characteristics.

SORT was originally designed for tracking pedestrians in surveillance video, and it uses first-order and second-order states to describe a pedestrian's position and speed using a standard kinematic model for the state transitions. A linear constant velocity model is used to propagate a target's identity into the next frame. This model is a good approximation when the camera is static and the pedestrian is walking. However, it is inappropriate for our first-person vision paradigm, as the head camera is often subject to slight movements due to a variety of unconscious behaviors. These movements create a velocity field across the entire video frame even when there is no significant, purposeful head motion. Therefore, a better approach is to model the camera motion between adjacent frames with the homography transformation estimated from each pair of frames. Given an estimated homography, it is easy to determine whether the camera motion corresponds to a significant amount of head motion. Moreover, since a homography is a global image transform, it can also be used to predict the motion of the bounding box as part of the smoothing filter for the object detections.

The state of each object is defined as the homogeneous coordinates of the top-left and

bottom-right points of its bounding box:

$$h = [x1, y1, z1, x2, y2, z2]$$

Then, the relationship between the states of a bounding box in adjacent frames can be modeled with the following equation:

$$h_t = diag(H_{t-1}, H_{t-1})h_{t-1},$$

where H_{t-1} is the homography matrix estimated from the frames at time t-1 and t, and the $diag$ operator constructs a block diagonal matrix containing two copies of the homography. H_{t-1} maps points from image I_{t-1} to image I_t. We use this relationship instead of the constant speed assumption to predict the state of bounding boxes in the next frame, as part of the Kalman filter.

The homography matrix is estimated by the matching feature points on two adjacent images. In this work, since the two images are adjacent frames in a video, we can utilize optical flow to find the matching features. We initialize feature points using the "good features to track" method [128] for every 5 frames, if the number of tracked feature points drops below a threshold (5000 in this work). In each frame, we calculate Lucas-Kanade optical flow [129] on the tracked feature points one step forward and then one step backward. If the resulting points still fall on the original points, we use these points to estimate the homography matrix. Note that in addition to its use in tracking TV screens, the homography also provides an estimate of the user's head motion, which is subsequently used to detect TV-watching behavior.

We follow the approach of [127] to link detected targets over time and fill in missing detections. Specifically, a target needs to be detected for T_{ini} frames to collect sufficient evidence that it corresponds to a new TV screen in the environment in order to prevent the accumulation of false positives. In addition, a track for a TV screen is discarded if it is not

detected for T_{lost} frames. In this work, we set $T_{ini} = 3$ and $T_{lost} = 6$. We will release our modified TV detection models and our version of the SORT method for TV tracking and make them freely available to the research community on our project website.

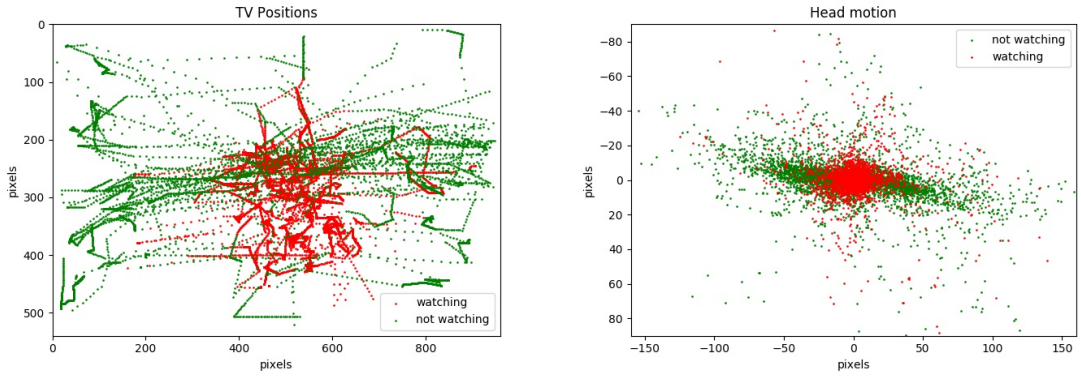
4.4 Attention Classifier

We now describe the inner analysis loop for our TV-watching detection system. The attention classifier labels each frame as corresponding to either watching or not watching a TV screen, and it can identify the specific location within the video frame containing the attention target (reducing the likelihood of obtaining the right answer for the wrong reason). We propose two features based on the location of the TV in the frame and the instantaneous camera motion. We then use a machine learning classification technique to predict TV watching using these features.

4.4.1 Feature Construction for Attention Classification

The information we use to predict attention comes from two sources: the static position of the TV and the head motion of the viewer. The intuition is that when one is engaged in watching TV, the TV is most likely in the center of the field of view and one's head is held still. We construct features to capture this intuition.

To verify that our feature-construction assumptions about TV watching are correct, we plot the TV positions and head motions for the moments in which people are and are not watching TV, respectively, in Figure 4.2a and Figure 4.2b using a subset of our labeled TV-watching data. We can see that when a viewer is watching TV, the TV is more likely to be located in the center of the frame, and the viewer's head tends to have relatively little movement in both the horizontal and vertical directions. This data provides partial confirmation of our hypothesis.



(a) TV positions (b) Head motions

Figure 4.2: Distribution of (a) TV positions and (b) head motions in the cases of watching TV (rd) and not watching TV (green).

Prior Distribution of Attention

Our attention classifier utilizes a model of the subject's attention to predict where they are statistically most likely to be looking during bouts of TV watching. In Section 4.5, we describe how the attention model is adapted over time to handle changes in screen-watching behavior. In this section, we describe how we obtain an initial model for the center prior from our existing dataset.

To obtain an initial model for the center prior, we construct a prior probability map from a subset of our labeled TV-watching data. In this dataset, we do not have explicit measurements of the point of gaze. However, we do have the position of the detected TV screens, and we further know the frames (via the data labeling) in which the subject was attending to the TV.

We fit a Gaussian distribution to the data of the TV positions in which the TV is being watched, *i.e.*, the red points in Figure 4.2a. The fitted Gaussian distribution is shown as an ellipse in Figure 4.3. There is slightly more variability in the vertical direction because the position of the TV varies with the head pose, the position of the nose bridge, and the subject's preference for wearing the video-recording glasses (*i.e.*, Pivothead). This distribution is used as the default attention distribution of the camera viewer before any

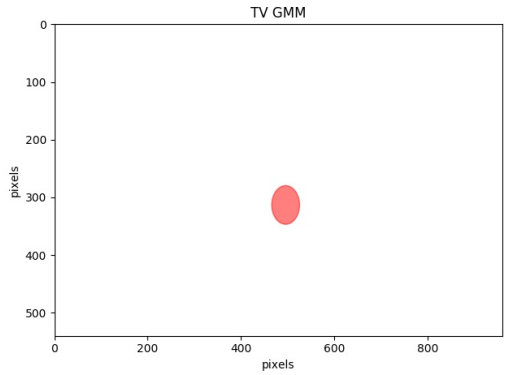


Figure 4.3: Prior attention distribution for TVs

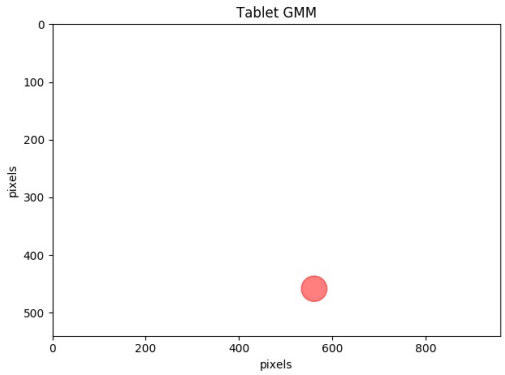


Figure 4.4: Prior attention distribution for tablets in the multidevice study

adaptation takes place.

The center prior model is used in two ways within our method. First, the mean (center position) is used to localize the attention distribution relative to any detected TV bounding box when constructing features for TV-watching detection. Second, the variance of the model is used as a criterion to identify whether the subject's attention pattern has shifted when adapting our model. Since the initial center prior estimate is obtained from videos recorded by different people, the initial model will have higher variance than we would expect from any single subject.

Head Motion

In order to determine whether the subject is keeping their head still, we require an estimate of their head motion over time. Since the wearable camera is rigidly mounted on the subject's head, this estimate can be obtained by computing the camera motion. The camera motion, in turn, can be obtained from the homography matrix, whose computation is described in Section 4.3.2. This two-dimensional camera motion vector constitutes the motion feature in Figure 4.1.

Feature Vector Formation

Given per-frame features consisting of smoothed bounding box positions for the detected TV screen, a constant center prior that provides the average fixation point for the subject, and the two-frame estimate of head motion, the next step is to construct a feature vector that can be used for classification. In order to gain robustness to noise, we aggregate features over a window containing the current frame and the five previous frames.

The first step is to combine the filtered TV screen track with the constant center prior to obtain a vector of six *attention probabilities* (one for each frame). These are computed by intersecting the center prior with the detected TV bounding box. The probability mass for the center prior which lies inside the bounding box is integrated to obtain the attention probability. If there is no bounding box for a TV-screen track for any frame, the probability is set to zero. Note that there can be multiple TV tracks at the same time in cases where multiple TV screens are detected over time. In such cases, we select the track with the highest attention probabilities as the candidate TV screen of interest. This strategy, in conjunction with our approach to adapting the center prior, allows the model to switch between TV screens when the user's attention shifts. The second step is to construct five 2-dimensional motion vectors from each pair of frames. The result is a 16-dimensional classification feature consisting of the six-dimensional attention probability and the 10-dimensional motion feature.

4.4.2 Machine Learning Model for Classification

Given the constructed feature vector described in Section 4.4.1, the next step is to design the machine learning model and associated temporal smoothing methods that are used for detecting TV screen-watching behavior. Our starting point is a random forest classifier for predicting attention to TV screens as a binary label based on the 16-dimensional feature vector constructed over a six-frame window. Random forests are very competitive classifiers for problems in which high-quality features are available, particularly in cases where

there is a limited amount of training data. One disadvantage of applying a random forest over a six-frame window is that the relatively short temporal scale results in noise, which is manifested as a lack of temporal smoothness (*i.e.*, flickering in the predicted labels). The lack of smoothness may not be significant for applications in which the primary variable is the amount of time spent watching TV screens. However, for applications that utilize the onset and offset of TV-watching bouts, the lack of temporal smoothness can be a significant problem. For example, such a lack would be an issue in trying to determine the preconditions for bouts of TV watching. We therefore describe two different smoothing methods that can be applied to the outputs of the random forest classifiers to improve the temporal continuity of the label predictions.

There are many ways to achieve temporal smoothness in an online fashion, including using a Kalman filter. A simple method is to use the strategy we used to create and delete trackers in TV detection. This approach removes the rapid changes in attention classification results. We can predict the onset and offset of a TV-watching event when there is sufficient evidence accumulated within a temporal window. Specifically, we label the onset of a TV-watching event if T_{ini} frames are labeled as watching and the offset of a TV-watching event if T_{lost} frames are labeled as not watching. We use $T_{ini} = 10$ and $T_{lost} = 30$ in this work, based on parameter tuning. This setting gives us a simple and computationally efficient method for online label smoothing.

As an alternative to the online approach, we developed a conditional random field (CRF) model [130] to take the temporal correlation of the predicted attention labels into consideration. The structure of the CRF, viewed as a graphical model, is shown in Figure 4.5. The model takes as input a sequence of probabilities of TV watching produced by a random forest regressor. Note that this is different from the online smoother, which simply accumulates per-frame detection outcomes. The CRF model penalizes the situation in which two adjacent frames are assigned different labels. This term in the energy model enforces temporal consistency and reduces the amount of flickering in the predicted label

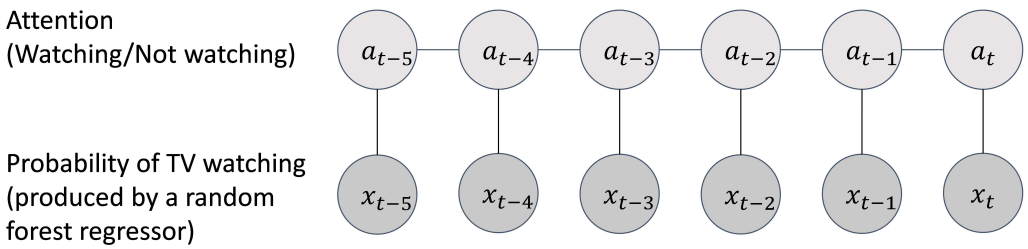


Figure 4.5: A CRF model is used to obtain smooth temporal predictions. The model takes as input a sequence of probabilities of TV watching produced by a random forest regressor. The CRF model penalizes the situation in which two adjacent frames are assigned different labels.

sequence.

4.5 Attention Model for Adapting the Center Prior

In Section 4.4, we described a classifier for detecting bouts of TV watching based on a static center prior model for the subject's attention. In practice, the center prior will vary from person to person and across conditions for the same person. Factors that effect the center prior include the location of the current TV screen in the environment, the subject's location and posture, and the position and orientation of the wearable video-recording glasses. For example, one difference arises when subjects are using tablets in their laps versus gazing at a mounted TV screen on the wall. Figure 4.4 shows the Gaussian distribution for the case of looking at a tablet. In contrast, Figure 4.3 gives the center distribution for a TV display on the wall. When people use handheld devices like tablets and phones, they usually look down, and their attention is focused on the lower part of the video frame. This is in contrast to looking at a display in the environment. The center of prior also depends on the height of the camera wearer's nose bridge. For people with a higher nose bridge, the TV will tend to appear in the lower part of the frame. In real applications, these factors along with others that we may not have encountered change the Gaussian distribution used to model attention. For example, for a person who is slouching on a couch watching a TV, the TV may appear in the lower part of the frame as compared to a person who sits upright.

Given all of these reasons why attention distribution can change, we adaptively estimate and update the center prior in our model. Given an initial Gaussian prior $X \sim N(\mu_0, \sigma_0)$, we re-estimate and adapt the mean and covariance over a sliding window of size of 30 seconds (i.e. $30 \times FPS$ frames). The mean and covariance can be updated with a constant time complexity that is independent of the window size. When the variance in the window is smaller than the variance of the original prior distribution, we use the mean of the Gaussian estimated in the window as the location of our attention distribution. This model accounts for the variations across participants and improves the attention classification results in our real-world experiments.

4.6 System Implementation

We now briefly describe the implementation of our system for TV-watching detection before discussing its experimental evaluation. The system has two parts in its current design. The first part is a method for recording video from subjects using off-the-shelf commercial head-worn camera systems. The second part is the software pipeline, implemented in Python,[2] that processes the videos and outputs the detected moments of TV watching. The video analysis is performed offline on a desktop with Intel Core i7-3930K CPU @ 3.20GHz x 12 and a NVIDIA TITAN X GPU. The object detection utilizes the GPU, while the rest of the algorithm runs on the CPU. While our current system is offline, our assessment of the computational requirements of the approach, detailed in this section, suggest that an online implementation is a feasible target for future work.

We use two commercial products to capture RGB video for our studies: The Pivothead SMART wearable camera is used in the experiments in Sections 4.7.3 and 4.7.4, while the SMI eye-tracking glasses are used to capture RGB video along with eye-tracking data for the experiments in Section 4.7.5. The Pivothead records video at a resolution of 1920x1080 at 30 fps. The camera is located in the bridge of a pair of glasses and can record video for

[2] All of the trained models and the software implementations for this project are available from the project website: `https://yunzhang07.github.io/tvwatch`.

approximately 50 minutes from a full charge of the integrated battery module provided by the manufacturer. We selected the Pivothead platform because it is lightweight and unobtrusive and has an acceptable battery life for our experimental purposes. The SMI glasses record video at a resolution of 1280x960 at 24fps. An outward-facing camera in the bridge of the glasses captures the scene, while eye trackers mounted in the frame track the participant's eye movements. The glasses are connected to a laptop during the experiments. The SMI is significantly more bulky and less comfortable to wear, but offers the advantage of providing eye-tracking data.

In training the TV-watching model described in Sections 4.4 and 4.5, we used data collected with the Pivothead, which constitute the majority of our collected videos. In order to apply the resulting model to the SMI videos during testing, we had to adjust the model parameters to account for the different aspect ratio and frame rate of the SMI platform. The attention prior was scaled according to the image size, and the motion vector was scaled by 0.8, based on the frame rate, before it was input to the prediction system. The ability to adapt the model to novel camera platforms is a strength of our approach, which results from our modular architecture.

We analyzed the computational cost of the algorithm to assess the feasibility of an online implementation. We benchmarked the runtime of different parts of our system on an 80-second video containing 2425 frames. Processing required 396 seconds, distributed as follows: 298 seconds are used to perform TV detection in each frame, and 96 seconds are used to compute the homography matrices for smoothing the detection results and constructing motion features. All other steps, including constructing features from the attention prior, predicting attention for single frames, and offline smoothing, require less than 2 seconds. Therefore, the effective frame rate for the offline implementation is 6 fps. In targeting an online implementation, several steps could be taken to save additional computation. One possibility is to utilize one of the more lightweight object detection network architectures such as SSD [131], which can process a 512x512 input video at 22 fps. In

addition, recent works in deep learning have demonstrated techniques to reduce the number of parameters and operations needed to make lightweight networks that can run on CPUs or even mobile devices, such as XNOR-Net [132], SqueezeNet [133], MobileNet [134], and ShuffleNet [135]. Using these methods, there is the potential to substantially reduce the time spent in TV detection, which constitutes 75% of the total computation time.

4.7 Experiments

We conducted four experimental evaluations of our method with two different subject populations. The first three studies utilized students recruited at Georgia Tech, while the fourth study utilized participants recruited from the greater Atlanta area. The four studies and their goals are as follows:

- *Feasibility Study* (Section 4.7.3) assessed the feasibility of video data collection, the accuracy of TV detection in Pivothead video, and the agreement between annotations produced by participants and independent raters.
- *Aware Home Study* (Section 4.7.4) examined TV-watching behavior in a living room environment with student participants.
- *Multi-Screen Study* (Section 4.7.5) evaluated the ability of the method to accommodate attention to multiple types of screens using a ground truth measure derived from eye tracking.
- *Naturalistic Study* (Section 4.7.6) evaluated the performance of the method in natural home environments drawn from a broad cross-section of the Atlanta population.

The Feasibility Study consists of 12 short videos recorded by four participants. It was used to construct the initial attention model and validate our evaluation approach. Specifically, we tested the inter-annotator reliability between the participant annotating their own first-person videos and the codes from independent raters. We also analyzed the distribution of

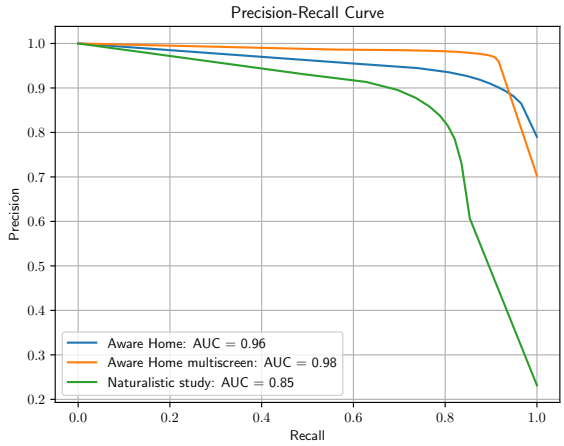


Figure 4.6: Precision-recall curve for the experiments in this paper.

attention and head movement and constructed the attention prior before evaluating our system's performance. Section 4.7.4 describes an experiment in a living room environment. Sixteen subjects had the ability to watch different types of TV programs in groups for an hour in the Aware Home lab on the Georgia Tech campus. Participants had the opportunity to eat and talk in addition to watching. We tested the performance of the model we trained using the feasibility study data on this dataset. The Multiscreen Study in Section 4.7.5 also took place in the Aware Home, but involved 10-minute sessions with multiple screens present and a protocol that encouraged participants to switch between screens. The Naturalistic Study in Section 4.7.6 recruited participants from the broader Atlanta population and obtained screen-watching examples from natural home environments. The precision-recall curves for detecting screen watching in each of these studies are compared in Figure 4.6. We will discuss these results in the subsections that follow. Ethical approval was obtained from the Georgia Institute of Technology ethics committee for all studies, and all participants provided written informed consent to participate in the studies.

We identified five primary sources of variability which can impact the performance of our method, and these dimensions informed our study designs:

- Variations in screen appearance

- Variations in watching behavior for different types of screens
- Subject-specific temporal variations in patterns of TV watching
- Postural and environmental variations
- Variations in the video-based assessment of TV watching by human raters

We now describe these sources of variability in more detail and briefly explain how our studies are designed to address them.

The starting point for our analysis of TV watching is the detection of screens, and systematic failures to identify screens would therefore negatively impact performance. Note that spurious failures (missing one or two frames in a sequence) will have negligible effect since we smooth the screen locations over time (see Section 4.3.2). Two sources of variability in screen appearance can effect the detector. The first is variations in viewpoint due to subjects watching from different vantage points. The second is variations in the appearance of TVs, resulting from different manufacturers and time periods. We addressed the issue of viewpoint variation in our Aware Home studies, as participants watched screens from a variety of positions. We addressed the issue of TV appearance variability by combining multiple existing image datasets to obtain a large and varied collection of labeled TV images, which was then used to develop and test the TV detector (see Section 4.3).

In addition to its performance on conventional televisions, our detector can also identify other types of screens such as monitors and tablets. The proliferation of devices with screens results in the second source of variability: the differences in watching behaviors for different types of screens. For example, while TVs tend to be watched close to eye level and from a distance, monitors are frequently viewed up close, and tablets and phones tend to be viewed at waist level. Our Multiscreen Study assessed these variations, and we obtained additional examples from our Naturalistic Study. A third issue is the variability in how participants allocate their attention to TV screens over time. Our Aware Home Study gave participants the opportunity to intersperse TV watching with eating and social

activity, and our Naturalistic Study provided additional datapoints from natural home environments. A related fourth issue is variability in posture, as individuals can watch TV while standing, lying down, slouched, etc. Our Naturalistic Study gave us an opportunity to assess this dimension of variation. A final issue concerns the ability to identify TV-watching ground truth labels using independent raters observing the recorded video. In our Feasibility Study, we compared the labels of independent raters to those of participants rating their own videos. In addition, by incorporating an eye tracker into our Multiscreen Study, we obtained additional objective corroborations of our ground truth ratings.

Given the importance of screens in people's lives and the extensive sources of variability in screen watching, the present study cannot be considered fully comprehensive or definitive. However, we feel that we have taken an important first step in demonstrating the feasibility of automated detection and identifying and characterizing some of the key sources of variability that impact performance. The section is organized as follows: Section 4.7.1 describes the performance metrics that are used across our four studies. The performance of the screen detector is evaluated in Section 4.7.2. Sections 4.7.3 through 4.7.6 present the results from the four studies.

4.7.1 Performance Metrics

The task of TV detection is an example of a standard object detection task in computer vision. We therefore utilize the performance measure known as the AP@0.5, which is closely related to the performance metric from the PASCAL VOC Object Detection Task [125].[3] The Average Precision (AP) is the area under the precision-recall curve that summarizes the TV detector's performance.

[3]This metric addresses the issue that an object in an image will tend to produce multiple detection outputs, because multiple bounding boxes that overlap significantly with the ground truth bounding box will tend to produce positive detections. AP@0.5 penalizes redundant detections by allowing the detection with the highest confidence to "claim" the ground truth bounding box as long as it overlaps spatially by a specified amount (0.5 in this case). All other detections are marked as false positives. This is important for our application, as we require one detected bounding box per occurrence of a screen in the image. Note that PASCAL VOC uses the related mAP measure, which is the mean of the AP over multiple classes.

In contrast to object detection, the classification of TV watching is evaluated by computing the agreement between predictions and ground truth over time. One approach would be to define events of TV watching and compute the intersection between detected events and the ground truth watching segments, analogous to what has been done in action detection [136, 37, 137]. However, in the use cases of interest to us, the primary goal is to measure the amount of time spent watching TV as accurately as possible, which can be captured by the frame-level metrics of precision and recall. In addition to focusing on the frames in which watching takes place, we also want to capture the system's performance in correctly rejecting frames that do not contain TV watching, so we use accuracy as an additional measure. In summary, the following complementary frame-level metrics are used to evaluate the performance of the TV-watching detector:

- **Precision:** Among the frames the system classifies as watching, precision gives the percentage that are correct.
- **Recall:** Among the frames with the ground truth label of watching, recall gives the percentage of frames that were correctly detected as watching by our method.
- **F1-score:** This is the geometric average of the precision and recall, which summarizes performance and facilitates comparisons between classifiers.
- **Accuracy:** This is the percentage of total frames correctly classified (as either watching or not watching).

In practice, applications of classification always involve a tradeoff between the two types of errors (false positives and negatives), and we use the precision-recall curve of our detectors to provide an overall performance summary (Figures 4.6 and 4.10 give examples).

In the following sections, we report the accuracy of the attention model on several different datasets. Here, we briefly explain the evaluation paradigm that we followed. We evaluated four different versions of our system. These different versions allow us to quantify the benefits of different aspects of our TV-watching model:

- The *random forest* model makes predictions for single frames without any temporal smoothing, and the attention prior is fixed and estimated from the video subset we used for the feasibility study.

- The *adaptive random forest* model includes the adaptation loop as discussed in Section 4.2

- The *online adaptive random forest* performs smoothing locally over the predicted labels by using a sliding window to smooth the predicted labels at the current frame using several subsequent frames. This approach ensures that we only change the predicted state of watching when the model has accumulated sufficient evidence.

- The *adaptive chain CRF* performs off-line smoothing over the predicted labels using a chain CRF model based on all frames from the video.

4.7.2 Screen Detection

Screen Dataset

The first step in our approach is to develop a screen detector. Our starting point was to leverage existing computer vision datasets that already contain labeled examples of TVs, monitors, and tablets, as illustrated in Figures 4.7 and 4.8. However, the labeling criteria used in these datasets is different from our goal, as an object is labeled if any part of it is visible in the image. These datasets therefore include TVs and monitors from a wide range of viewing angles and scales and with different levels of occlusion. In contrast, our screen-viewing application greatly restricts the range of feasible viewing angles and scales. Moreover, the fact that our videos are collected from head-worn cameras may result in additional variations in viewpoint, scale, and appearance, which are not covered in standard datasets. We addressed this issue by constructing an additional dataset containing screens under suitable watching conditions. In particular, we exclude images showing the back of

Figure 4.7: Examples of TV images in the PASCAL VOC dataset

Figure 4.8: Examples of TV images in MS COCO dataset

Figure 4.9: Examples of screen images in our new screen dataset

the devices, as well as screens that are far away or heavily occluded. Figure 4.9 shows some of the example images in our screen dataset.

Our new screen dataset includes *TV and monitor* bounding box annotations for 500 images and 60 video clips downloaded from YouTube and *tablet* annotations for 500 images downloaded from Google Images. The selection of images and clips covers a variety of different types of screens that are occluded by less than 30%. Screens which are partially occluded are labeled as such. We used the video annotation tool *vatic* to annotate the video clips. Note that although we have different labels for TVs, monitors, and hand-held devices, it is sometimes difficult to make a clear distinction between the classes. A tablet placed on a stand could appear to be similar to a monitor and could serve the same purpose. In practice, we find that the detector is able to detect screens reliably overall but can be confused about the type of the screen, particularly when it is partially occluded. We therefore combine the screen classes into a single screen category for the purpose of detecting screen watching.

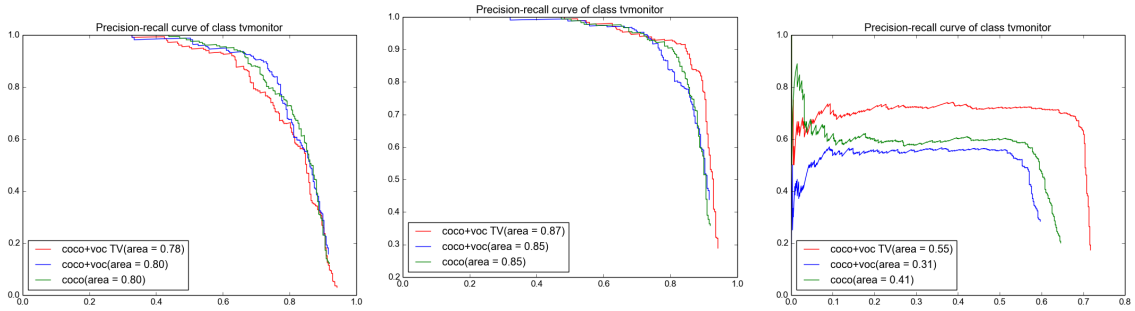
Detection Network

Our screen-detection approach is based on Faster RCNN [124] and uses the VGG16 network architecture because it achieves satisfactory results and consumes moderate computational power. We took the model from [124], which is pretrained on the ImageNet dataset, and then fine-tuned it using the COCO training set. The fine-tuned model on COCO was obtained from [124]. Subsequently, we fine-tuned it further on the PASCAL VOC 2007 trainval set and PASCAL VOC 2012 trainval set following the settings from [124], and we continued to fine-tune it on TV images from the COCO and PASCAL VOC datasets with a learning rate of 0.0003 for 8000 iterations and 0.00003 for 2000 iterations. We tested the network on the PASCAL VOC 2007 test dataset (07test), images with TVs in the VOC 2007 test dataset (07testTV), and TV and monitor images in the screen dataset (screenTV). The 07testTV and screenTV datasets only have images with TVs. The accuracy of the detectors is shown in Table 4.1, which demonstrates that fine-tuning on TV images decreases the overall performance but increases the performance on TV images. There are two potential factors at work. First, the region proposal network (RPN) of the original Faster RCNN network generates generic object proposals for all classes. When we fine-tune the network for the TV class, the RPN may generate proposals that are biased for TV objects. Therefore, the network rejects more non-TV regions at the RPN stage. Second, the network may be exploiting scene context in learning to detect TVs, as they frequently occur indoors and in conjunction with certain types of furniture. As the network adapts to context in the TV images, the performance for other categories can be impacted. Since we only require accurate screen detection, this biasing is not a problem.

We examined the precision-recall curve for screen detection using three different datasets to gain greater insight into the performance of the model, as illustrated in Figure 4.10. The model trained on COCO+VOC TV images has the highest performance on TV detection, shown in Figures 4.10(b) and (c), but the worst performance on VOC 2007, as shown in Figure 4.10(a). This result indicates that the model is specializing to the screen-detection

Table 4.1: TV detection accuracy

Model	*Training Set*	*07test*	*07testTV*	*screenTV*
1	coco	0.7976	0.8493	0.4113
2	coco+07+12	0.7993	0.8468	0.3083
3	coco+07+12+TV	0.7808	0.8744	0.5480

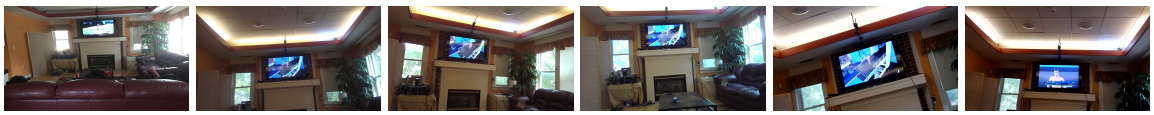


(a) VOC 2007 test

(b) VOC 2007 test, images with TV

(c) Youtube TV

Figure 4.10: Precision-recall curve for TV detectors



(a) (b) (c) (d) (e) (f)

Figure 4.11: Aware Home Study: (a) Picture of the living room of the Aware Home lab, in which the study was conducted. (b)-(f) depict sample video frames collected during the study.

(a) (b) (c) (d) (e) (f)

Figure 4.12: Multiscreen Study: Sample video frames from the Aware Home study involving attention to multiple types of screens. Screens with red bounding boxes are identified as the target of the subject's gaze. The green circle in (b) is the measured point of gaze produced by the eye tracker.

task at the expense of the other categories.

4.7.3 Feasibility Study

A critical issue in our experiments is how to obtain a ground truth label for screen watching on a frame-by-frame basis. The participant who wore the camera is in perhaps the best position to review the videos and assign labels to the frames, but they may not always be willing or able to conduct this burdensome task. Independent annotators in the form of undergraduate research assistants (RAs) are available to label such videos and can greatly facilitate scaling the dataset.

To establish the validity of annotations produced by independent raters, we used a small subset of videos we collected to conduct a feasibility study. We tested the agreement between the annotations produced by the participant and those produced by independent raters. Participants recorded short video clips using the Pivothead SMART wearable cameras. Participants sat at different locations of their own choosing in this experiment, and the TV is therefore of different sizes and poses within the wearable camera frames. We collected 12 1-2 min sessions from four subjects. Subjects in these videos typically shift their attention every 5-30 seconds and spend approximately half their session watching the screen and the other half looking away. This data was used to develop an initial attention model and validate our annotation approach.

For the purpose of validation, each video was annotated with TV bounding boxes, and labels for watching or not watching TV were assigned to each frame. One set of labels was provided by the participant and the other by independent raters. Each of the four participants annotated their own videos and also played the role of an independent rater by annotating the video from one of the other participants, thereby enabling us to measure inter-rater reliability. We used Cohen's Kappa score to measure the agreement among the annotators. The average Kappa score was 0.83, which is a satisfactory level of agreement. We conclude that independent raters can effectively score the first-person videos of participants and assign labels for TV watching that are comparable to the labels that the subjects themselves would assign.

Table 4.2: Attention prediction for Aware Home TV-watching dataset

Model	*Precision*	*Recall*	*F1*	*Accuracy*
Random forest	0.9197	0.8210	0.8675	0.8019
Adaptive random forest	0.9185	0.8755	0.8964	0.8402
Online adaptive random forest	0.9169	0.9451	0.9256	0.8800
Adaptive chain CRF	0.9141	0.8958	0.9049	0.8512

There are a few caveats worthy of additional discussion. First, the sample of videos in the feasibility study was somewhat limited, and it is possible that higher levels of disagreement could arise in more complex TV-watching situations. Furthermore, there is an element of subjectivity in TV watching that cannot be completely avoided. Is a participant who is staring at a screen really watching it, or might they be daydreaming or asleep? In this regard, the relatively narrow field of view of the Pivothead glasses is actually somewhat advantageous. Maintaining a screen in the field of view requires significant head control, and when a participant is asleep or distracted it is less likely that an unwatched screen will be in view. In contrast, a camera with a 360-degree field of view would see all the screens in the room at all times, and if the participant kept their head still it might be less clear which screen (if any) they are viewing. Note that for use cases related to sedentary behavior, knowing that a participant was stationary for long periods of time with one or more screens in view is still useful data, even if their detailed pattern of attention is not perfectly clear. Note also that future work could potentially include other physiological sensors along with the camera data to facilitate the assessment of the subject's patterns of attention.

4.7.4 Aware Home TV Watching

Experiment Setting

This study took place in a living room with a couch and chairs in front of a conventional TV set. The room is located in the Aware Home lab at Georgia Tech, a laboratory space contained within a normal house to support smart home studies. We instructed our partici-

pants to watch a movie, a TV show, or a sporting event in groups of 2-5 people. Snacks and drinks were provided. The subjects were informed that they could eat, drink and chat during the session. This protocol was designed to elicit naturalistic TV-watching behavior, as the participant's attention shifted between the screen and other objects and people in their environment. Fifteen subjects were recruited, and 13 hours of video were recorded. The videos recorded from each subject varied in length from 30 minutes to 1 hour. One of the subjects attended two sessions. Seven of the subjects were males and eight were females. All participants were between 18 to 32 years of age. Fourteen of them were students at the graduate or undergraduate level, and one of them was a recent college graduate.

Ground truth labels for TV watching were provided by independent annotators, and spot-checking was used to ensure that the labels were valid. Figure 4.11a illustrates the experimental setting, and Figures 4.11b – 4.11f are sample frames from videos recorded by five subjects sitting at different locations within the room.

To assess the participants' opinions about the experiment, we administered a questionnaire at the end of the study. The question 'How similar was the experience of watching TV in this study to a typical TV-watching experience in your daily life?' received an average response of 5.85 on a scale of 1 to 7, where 1 is "the least similar" and 7 is "the most similar". The primary differences reported by the subjects were the need to wear the Pivothead glasses, the TV volume, and other personal preferences. When asked to name similarities, the subjects reported that the environment was relaxing and comfortable, there were snacks and drinks, and watching was interspersed with talking.

TV Watching Prediction

We evaluated the accuracy of the attention model on each frame of the recorded videos from the Aware Home study. The results are shown in Table 4.2. The first row in the table shows the results using the single-frame random forest model. In this experiment, the prior attention distribution is assumed to be fixed throughout the experiment. The second row

shows the results using the same model but using an adaptive attention distribution. The adaptive attention distribution model improves the attention classification. Row 3 shows the results for local smoothing. Row 4 shows the results for the CRF offline model. Rows 3 and 4 results are the result of additional smoothing beyond the results of row 2. The model used in this experiment was trained on a subset of data (from the feasibility study). We did not retrain the model on this dataset. The results indicate that the model is able to generalize to unseen examples.

The smoothing methods improve the results because this dataset has long events, and these models can filter out short predictions and connect nearby predictions. The offline smoothing model outperforms the results in row 2 but does not perform better than online smoothing. This result is not particularly surprising, because our online model uses the T_t frames ahead of the current frame to predict the attention, and thus, the prediction lags T_t frames. We also calculated the accuracy, given in the last column of the table, as the percentage of frames assigned the correct predicted label, as this measure has been used in other studies [37].

Figure 4.13 shows the precision-recall curve for different models on this dataset. The accuracy of the online smoothing model is 88.00%. The subjects watched TV for 79.02% of their time. Therefore, a naive estimator can achieve an accuracy of 79.02%. In comparison, the estimator in [37] achieves an accuracy of 80.3% using viewer-facing color and depth cameras. They use a dataset collected in a home environment. Given the differences in our approach and evaluation setting, the results are not directly comparable. In particular, their system is installed on the TV and evaluated on a different dataset.

4.7.5 Multiscreen Study with Eye-Tracking Validation

In this study, subjects were instructed to use multiple devices while wearing a pair of SMI eye-tracking glasses. During the experiment, the glasses were tethered to, and powered by, a laptop. The experiment was conducted in the same Aware Home lab used in the

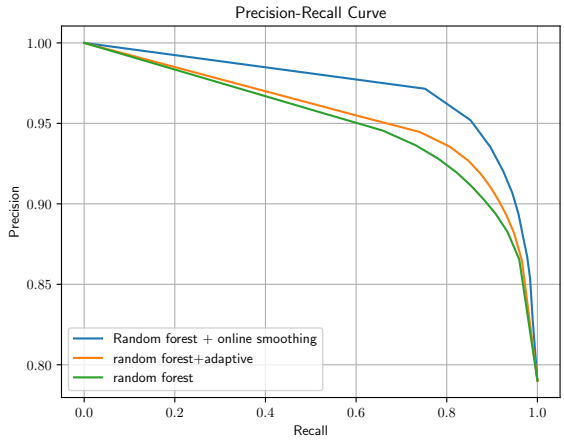


Figure 4.13: Precision-recall curves for different models in the Aware Home study

experiments from Section 4.7.4. Each session involved a single subject, who was asked to watch a TV documentary. While they were watching the screen, we sent a message to their phone or a tablet that prompted them to perform two tasks. First, they had to reply to an email on the handheld device. Second, we sent a message prompting them to read two pages of a book. One of two documentaries was shown to each subject. Each session lasted 10-15 minutes. We recruited eight subjects for this study. All participants were undergraduate or graduate level students between 19 to 36 years old. Seven of the participants were male and one was female. Figure 4.12 shows some example frames from the videos recorded in this experiment.

We administrated the same questionnaire as in Section 4.7.4. The average score provided in response to the question 'How similar was the experience of watching TV in this study to a typical TV-watching experience in your daily life?' in this experiment was 4.5, lower than the responses we received in Section 4.7.4. This is in part because the eye-tracking glasses used in this experiment were tethered to a laptop, so the subject could not move freely. In addition, we gave the participants instructions via text message, which made them feel like they were being "supervised" by the experimenter.

This study was designed to achieve four goals. First, we wanted to obtain additional validation of our annotation approach by comparing the human annotations from RGB

videos to those based on eye-tracking data. Second, we tested our attention model in a scenario that involved multiple types of devices with screens (TV set, tablet and phone) that appeared at different locations and distances relative to the subjects. Third, we evaluated the ability of our RGB image-based attention model to localize the device which was being gazed at by comparing our results to those obtained with eye tracking. Finally, this study investigated the feasibility of using automatically detected screens in conjunction with eye-tracking data to study TV-watching behavior. In the future, if eye trackers become smaller, lighter in weight, and lower in power consumption, they may eventually provide a feasible alternative to video recorders. Even in that case, however, it will be necessary to automatically detect the gaze targets in order to correctly interpret the gaze data. Our results show that our screen detection algorithm can be used to reliably identify TV screens as gaze targets in an eye-tracking application. We used the same attention model as in Section 4.7.4 to process all of the multiscreen videos.

Annotation Validation

We conducted a study with two steps to obtain additional validation for our annotation approach based on independent raters. In the first step, the raters annotated the RGB videos from the multiscreen experiment, in exactly same manner as the Aware Home study, by marking the start and end of each period in which the participant was watching a screen. The raters also identified the type of screen by coding each frame with one of the following codes: $\{$TV, tablet, phone, no screen in use$\}$. Note: for the purpose of detecting screen watching, we grouped the categories of TV, tablet, and phone together into a single "TV" class. This was done to simplify the experiment. The prediction of when the subject is attending to specific types of screens is left as a future research topic. In the second step, the raters made an additional pass over all of the videos, during which the gaze-tracking data from the SMI glasses was superimposed on each frame (see Figure 4.12b for an example frame). Comparing the ratings obtained with and without eye tracking allows us to assess

the extent to which access to eye-tracking data will change the assessment of TV watching.

One caveat is that the eye-tracking data obtained during the study was not perfect for two reasons. 1) The tracked gaze points are not smooth. During a watching event, the point of gaze is located primarily inside the target but can have abrupt jumps that deviate from the main path. 2) The eye-tracking data is only accurate for objects located at the distance for which the camera has been calibrated. In our study, the TV and the hand-held devices are located at drastically different distances from the subject. The eye tracker was calibrated for the distance to the TV, and as a result does not produce accurate tracking data for handheld devices. Given the relatively smaller screens of handheld devices, the eye-tracking points may not lie on the device when the person is looking at it. We will discuss this limitation in the following sections. In the current section, the eye-tracking based annotation is left to the annotator's judgment when she examines the videos with eye-tracking points superimposed. For example, a frame could be labeled as "looking at the phone" when the point of gaze is located outside the phone.

The annotations obtained from videos with and without eye-tracking data superimposed agree in 98.62% of the frames. This agreement provides additional validation for the annotations approach used throughout our experiments. For the remainder of this section, we use the annotations obtained from videos with eye-tracking data superimposed as the ground truth.

Screen Use Detection with Eye-Tracking Camera

In this experiment, we explore the possibility of using eye-tracking data to automatically detect screen-watching events by combining it with the TV detector.

For each frame, we classify behavior as "watching" when the gaze-tracking point is inside any detected screen and as "not watching" otherwise. We obtain a precision of 97.92%, recall of 80.00% and overall accuracy of 84.74% with this approach. In this dataset, the subject is looking at one of the devices 70.29% of the time. The errors come

from two sources: errors in screen detection and those in eye-tracking results. From the visualization of the results, we believe that detection error is the dominant factor. However, a quantitative analysis is not available unless we annotate the bounding boxes of screens on each frame. The high precision and low recall indicates that miss detection of screens happens more often than false positives.

Figures 4.12a – 4.12c show some failure modes. Figure 4.12a provides an example in which the TV is not detected because the screen is mostly dark and looks like it is turned off. This situation happened frequently in this experiment because one of the videos we showed consists of approximately 1 minute of such content. Figure 4.12b is an example where the gaze point is outside of the phone the person is using because the camera is not calibrated for the distance of hand-held devices. Figure 4.12c is discussed in the next section.

Screen Use Detection without Eye-Tracking Data

We tested our attention modeling system on videos without eye-tracking points. We classify each frame as "watching" and "not watching" as in the previous experiments. Furthermore, we check whether the bounding box selected by our attention system is the same as that selected by eye-tracking data. In this way, we can test our model's ability to localize the device of interest.

For the binary classification task, we obtain a precision of 98.13%, a recall of 87.07% and an accuracy of 89.74% with the adaptive, online smoothing model. The numbers are higher than those using eye-tracking data for two reasons. First, here we may predict the right label for the wrong reason. In an image like Figure 4.12c, the TV might not be detected, but the fireplace is detected as a TV and classified as being watched. Second, the eye-tracking results are not accurate. For images like 4.12b, the prediction using eye-tracking could be wrong. Based on the visualization of the results, the first type of error dominates. To assess the system, we also check the accuracy of TV localization. Among

Table 4.3: Attention prediction for naturalistic dataset

Model	*Precision*	*Recall*	*F1*	*Accuracy*
Random forest	0.8750	0.7080	0.7827	0.9090
Adaptive random forest	0.8585	0.7654	0.8093	0.9165
Online adaptive random forest	0.8714	0.8168	0.8432	0.9297
Adaptive chain CRF	0.8555	0.7873	0.8200	0.9200

those frames that are labeled as "watching" by the eye-tracking points and our attention model, 80.33% of devices are localized by our attention model to the same detections found by eye-tracking points. The adaptive model improved this number from 73.28%. This evaluation rules out the first type of error but not the second. Figures 4.12d, 4.12e and 4.12f are correct detections on TV, phone and tablet.

4.7.6 Naturalistic Study

To test the performance of our system in a naturalistic scenario, we recruited an additional cohort from a broader population in Atlanta instead of university students. We asked the subjects to wear the Pivothead glasses and record aspects of their daily life when they were not at work and felt comfortable doing so. Participants were allowed to record in their homes if they lived alone, or if all occupants had consented to sharing and there were no children present. Our instructions to the participants did not mention any tasks or activities related to screen use, and the collected videos cover both indoor and outdoor activities of significant variety. Participants were allowed to review their recorded videos and delete either entire videos or parts of videos before transferring them to us. We collected 98.0 hours of videos from 8 subjects. Each subject provided from 1.87 to 39.82 hours of videos, with a median length of 7.77 hours. The participants were between 22 to 67 years old. Four participants were female and four male. Five participants were employed full-time at intake, two were employed part-time, and one was retired. The convenience sample did not include any students.

Table 4.3 gives the results from a subset of the data consisting of 31.6 hours of videos from 8 participants. We sampled the videos randomly within each subject and normalized the lengths of the video clips across the participants. On average, the participants in this footage were using a screen for 25% of the time. As in previous sections, the attention model from the feasibility study was used to analyze the data. The accuracy of our screen watching detection system on these videos is 92.97%. Figure 4.6 shows that the performance on this dataset is lower than it was on our other datasets which were collected in laboratory settings. This difference is not surprising, as the naturalistic dataset includes a wider range of postures and approaches to screen viewing in comparison to more structured experiments. These results demonstrate that our system works well even in real-world scenarios in which the types of screens the participants can encounter and the types of activities that they can perform are not specified in the study design.

4.8 Discussion

This paper introduces a novel approach to detecting an individual's screen-watching activities by analyzing video recorded by a wearable camera embedded in a pair of glasses. Our approach is based on an attention model for screen viewing which utilizes two features to capture information about the location of the screen and the head motion of the screen viewer. We have demonstrated that these two features are discriminative for classifying attention to screens. We developed a series of classifier designs that explored different approaches to temporal smoothing in classifying screen watching. We compared their performance in four different studies involving two study populations. We conducted structured examinations of screen-watching behavior from student participants in the Aware Home research lab at Georgia Tech. We obtained an additional naturalistic sample of videos from a population of nonstudents drawn from the greater Atlanta area. Our experiments demonstrate that our analysis approach can obtain good results over a broad range of screen-watching contexts and behaviors. Our attention model, software, and structured

TV-watching datasets are available to the research community free of charge. We hope this work will spur additional interest in automated approaches to assessing screen watching and promote the use of wearable cameras in mobile health [138] research.

The key innovation in our approach is an attention model that allows us to estimate the participant's attention to screens *without* having the ability to observe or track their eye movements. This utility is important in practice because video recorders are much more affordable and usable than wearable eye trackers. Although the current system achieves good performance, there are some observed challenges and limitations of the system that are worth discussing. One challenge is the limited field of view (FOV) of the video recorders currently on the market, in comparison to the human FOV. While this limitation can be beneficial in reducing false positives coming from unattended background screens in the environment, it also means that the participant could be gazing at a screen that is not visible to the camera. When people use tablets and phones, for example, they tend to place the devices at a position well below their head, which takes them out of the camera frame. The same problem can arise when a participant looks up at a screen, as illustrated in Figure 4.12c. In these scenarios, a human annotator might be able to infer the attention to a screen from the context. This could be an interesting direction for future work.

A second limitation of our current attention model is the inability to reliably detect which screen the participant is looking at under all conditions. For example, when there are multiple devices in view and people switch their attention between different screens without moving their head, our system is unable to detect the shift in gaze. Another example arises when a TV is in the frame but the participant is paying attention to something that is close to the TV, and our method cannot resolve the difference. Despite these limitations, our system is a first, successful attempt at the automated detection of screen watching, and it provides a new tool for the quantitative assessment of screen-watching behavior, which can complement conventional questionnaire-based and human annotation-based methods. Our algorithm-based approach could also serve as a first step before human annotation to

improve annotation efficiency.

We believe that this work can also provide a starting point for a variety of health applications. Our approach could be used to shed light on how bouts of attention to screens are structured in time, and to identify the conditions under which multitasking to multiple screens occurs. The ability to record video before, after, and during bouts of TV watching could illuminate how this pervasive activity is integrated into other activities of daily living. Our system can provide a first step towards assessing exposure to advertising content through additional analysis of the types of screen content to which participants are attending. In the future, when more powerful hardware platforms with longer battery lifetimes become available, it may be possible to detect screen watching in real time and provide interventions to bring about a change in screen-watching behavior. We hope that this work can provide a foundation for future studies of this important behavior using automated data collection and analysis methods.

CHAPTER 5
SYNCWISE: WINDOW INDUCED SHIFT ESTIMATION FOR SYNCHRONIZATION OF VIDEO AND ACCELEROMETRY FROM WEARABLE SENSORS

5.1 Introduction

The temporally precise annotation of sensor data is necessary in order to build models that can passively sense and infer behavior from sensor signals. For behaviors involving limb movements, such as smoking, eating, and brushing, video recordings from wearable cameras are increasingly being used to obtain temporally precise ground truth labels.

The cameras are worn and positioned to capture movements of interest under field conditions (see Fig. 5.1). Video is recorded simultaneously with the target mobile sensor data, and standard video coding is used to obtain ground truth labels for the sensor streams. These data can be used both to validate the accuracy of existing methods and to train new models. This approach has been used for eating, drinking, and brushing activities [139, 85, 86, 140, 87, 76] and is particularly valuable for fine-grained activities lasting on the order of seconds. However, this approach requires *accurate time synchronization* between the video sequence and the sensor data streams so that annotations obtained from video can be automatically transferred to label the sensor data. Any temporal misalignment between the video and sensor streams will result in label noise (i.e., incorrect labeling of the sensor data) and can significantly degrade the accuracy of the detector.

Because it is common for commodity sensor hardware to utilize independent, unsychronized clocks, previous sensor system architectures have incorporated effective approaches to sensor synchronization [73, 62, 75]. Unfortunately, these approaches cannot be easily extended to video capture. In contrast to sensor network approaches [73], commercially

available wearable video cameras such as GoPro are not designed for synchronization with other noncamera sensors.[1]

In addition, battery constraints make it infeasible to transmit video data wirelessly so that it cannot be time-stamped at a central collection point simultaneously with other sensor streams [75]. As a result of these issues, the problem of time synchronization of video cameras [67, 66, 142] and wearable devices [89] is well-known within the community to be a practical challenge in study implementation and a silent killer of data accuracy [139].[2]

When data is collected under laboratory conditions, many strategies can be used to establish synchronization points, such as the well-known clapperboard for audio-visual (AV) synchronization or the use of special hand gestures to synchronize body-worn accelerometers with cameras [89, 80]. These approaches are impractical for field studies as they impose significant burden on participants and rely on their adherence [144]. Alternatively, manual synchronization can be performed with tools such as ELAN [83] or Chronoviz [84]. This approach is laborious and time-consuming, and as a result of clock drift, it may need to be performed at multiple time points across a long recording.[3] It follows that there is a need for a flexible, general-purpose solution for synchronizing cameras with other mobile sensors that can be applied to field-collected data, does not impose any additional burden on participants, and is fully automatic.

In this work, we introduce a fully automatic method called Window Induced Shift Estimation for Synchronization of video and accelerometry (SyncWISE). Given a clip of video and accelerometry data, it outputs the time offset for synchronization (see Fig. 5.1(a)). We address the two key technical challenges of *partial observability* and *coordinate registration*. Partial observability refers to the fact that the time intervals in which synchronization

[1]While add-on products from third-party vendors, such as SyncBac Pro and ':pulse' from Timecode Systems [141], can provide synchronization solutions for wearable cameras, they are limited to synchronizing multiple cameras and do not address our scenario.

[2]Even something as basic as synchronizing audio and video for film-making has a long history of challenges and failures. For example, a legendary live performance by Aretha Franklin in 1972 was not released for 46 years, due in part to the failure to synchronize the video and audio properly during recording [143].

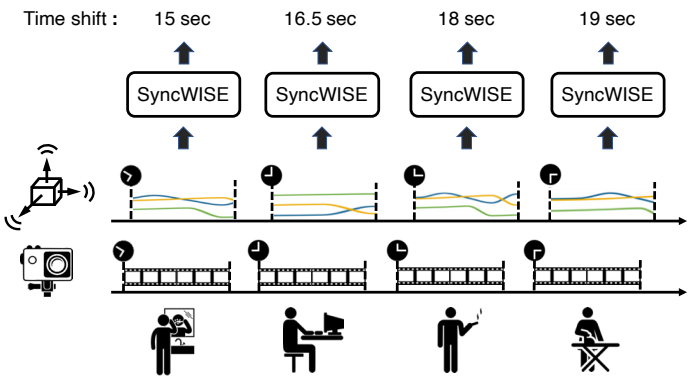
[3]Several authors have investigated clock drift arising in video cameras [64, 145]. The GoPro has an average drift of 1 second per hour, which is roughly the same duration as many fine-grained gestures.

points can be reliably identified are sporadically distributed. For example, a chest-worn camera on a participant standing at a street corner will capture significant dynamic video content, while a co-located accelerometer will register no movement. This is in contrast to prior work [63, 69] which has implicitly assumed that all moments of time are equally good for estimating synchronization. We address partial observability via a kernel density estimation approach in which weighted segment pairs are correlated and their votes aggregated to obtain the final offset. The second challenge of coordinate registration arises in synchronizing video with motion-based sensors, such as accelerometers, that output their data with respect to a 3D coordinate system. In this case, the correct comparison of the signals requires the two coordinate systems (camera and sensor) to be registered, so that corresponding directions of movement are being compared. In contrast, prior work on sensor synchronization in autonomous vehicles [63] leverages the fact that sensors are rigidly mounted and calibrated during installation. We address coordinate registration by using a PCA analysis to identify a common principal direction between modalities prior to registration. We validate our SyncWISE method on two datasets: the CMU-MMAC activity dataset [77], and a novel real-world dataset, called Sense2StopSync *(S2S-Sync)*, from a smoking cessation field study with 21 participants, consisting of 45.2 hours of recordings over the three days prior to their quitting. This work makes the following three contributions:

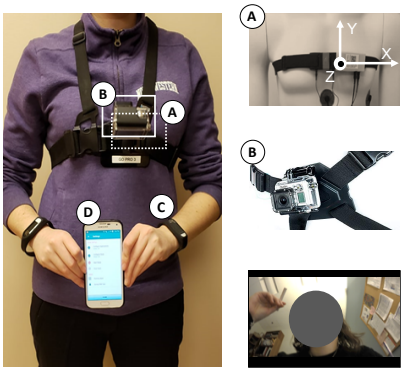
- We introduce the *SyncWISE* method for automatically synchronizing video clips with motion-based sensor data such as accelerometers and inertial measurement units (IMUs). We believe we are the first to identify and address the challenges of partial observability and coordinate registration that arise in the field environment.

- We provide the novel Sense2StopSync *(S2S-Sync)* dataset to the research community,[4] comprising 45.2 hours of time-synchronized optical-flow videos and accelerom-

[4]The dataset and code are available from the project website: `https://yunzhang07.github.io/SyncWISE/`



(a)



(b)

Figure 5.1: (a) Illustration of input and output in our SyncWISE system. (b) The wearable sensory platform consists of (A) a chest-worn sensor suite containing a 3-axis accelerometer worn underneath the clothes; (B) a GoPro video camera (an example of video camera footage is provided below the camera); (C) a wrist-worn sensor containing a 3-axis accelerometer and a 3-axis gyroscope worn on both wrists; and (D) a study smartphone with data-logging software.

etery data from two chest-worn devices collected from 21 subjects with annotations of smoking and feeding gestures.

- We present state-of-the-art automatic synchronization results for the CMU-MMAC and S2S-Sync datasets that significantly outperform two versions of a baseline method [63]. The software will be made freely available.

5.2 Study Design and Data Collection

We now detail the collection of the S2S-Sync dataset. We first describe the study design in Sec. 5.2.1. The sensor data collection is outlined in Sec. 5.2.2, and the approach to data annotation is described in Sec. 5.2.3.

5.2.1 Study Design

Data was collected during a smoking cessation study, *Sense2Stop*. Participants (age 18-65) were eligible for Sense2Stop if they had smoked at least 1 cigarette per day for the past year. The S2S-Sync dataset is generated from Sense2Stop during the three-day pre-quitting period in which subjects exhibited maintenance behavior (*i.e.*, typical smoking patterns). The pre-quitting period provided baseline data for participants' smoking and eating behaviors. The video collected during pre-quitting supports the annotation of smoking and eating behaviors to validate and refine machine-learned models.

Study Timeline

On Day 1, participants visited the lab, where they were fitted with the mobile devices and received instructions. The pre-quitting phase ended on Day 4, when participants returned to the lab to upload their wearable video data to the study servers. During this visit, participants had the opportunity to delete any video footage that they did not wish to share. Participants then continued into the post-quitting period without the video camera.

Participant Instructions

During the pre-quitting period, participants were instructed to wear the provided GoPro camera for four hours on at least two separate days that included at least one smoking event and the eating of at least one meal and one snack, for a total of eight hours of in-the-wild recorded video.

5.2.2 Devices

The wearable devices worn by the participants included a GoPro video camera strapped to their chest, a chest-worn sensor suite comprising an accelerometer, electrocardiography (ECG) sensor and respiratory plethysmography (RIP) sensor, and a pair of wrist-worn devices with triaxial accelerometers and gyroscopes on each wrist. Additionally, participants were provided with a study-dedicated smartphone with data collection software installed. We focus our analysis on synchronizing the chest-worn accelerometer and GoPro video camera.

Video Camera

Participants wore a GoPro Hero 4 camera recording 1080p video at 30 Hz. The GoPro was mounted on the chest with a chest-mount strap and case that protects the camera and image quality from dust, water, and other elements. The camera was oriented towards the participant's face. The captured video was stored on a µSD card as a series of MP4 files, each of which is 4 GB and 17 minutes and 43 seconds long. Before deployment, the GoPro's clock was synchronized with a PC to the National Institute of Standards and Technology (NIST) time server. As an added precaution, the camera was briefly oriented towards the PC to record the NIST time webpage (`time.gov`), providing an additional sync reference before the camera goes out into the field.

Accelerometers and Data Logging

Two sets of accelerometers were used in our study. The accelerometer from the chest-worn device, AutoSense [146], sampled at 10.66 Hz, was used in all of our automatic synchronization experiments. Note that this device is mounted on a harness that is separate from the GoPro. In Fig. 5.1(b), the accelerometer is on (A) while the camera is on (B). Thus, while the two devices are roughly co-located, the camera is capable of significant movement relative to the accelerometer, including changes to its orientation. Additional accelerometers

from the MotionSense wristband [147], mounted one on each wrist and sampled at 16 Hz, were used by the annotators during manual synchronization (see Sec. 5.2.3).[5]

Data from all accelerometers was transmitted to the study phone wirelessly and logged on an encrypted μSD card by the open-source mCerebrum smartphone app [75]. Then the data were periodically uploaded to a secure server running the open-source Cerebral Cortex [148]. The mCerebrum app time-stamps each packet of data to a common clock, thereby synchronizing the accelerometry signals to each other. Wireless transmission can result in dropped packets and packets arriving out of order. The software performs interpolation for small gaps in the data, but significant gaps in the accelerometry signal remain. Our approach to synchronization explicitly accounts for missing data and poor signal quality, which are endemic to mHealth applications [149, 150].

5.2.3 Data Screening and Annotation

The unit of data for our experiments is a video clip with an associated segment of accelerometry data. Each clip corresponds to one MP4 file captured by the GoPro, with a maximum duration of 17 minutes and 43 seconds. Clips that were shorter than 30 seconds were discarded, resulting in 378 clips from 34 participants. The timestamps recorded by the GoPro camera are used to identify the segment of accelerometry data which is paired with the clip. This is an extremely crude correspondence with substantial error. Additional screening steps, detailed in Sec. 5.2.3, resulted in the final S2S-Sync dataset comprising 163 video clips of 45.2 hours in total duration with associated accelerometry data from a total of 21 participants. Each participant contributed an average of 2.15 hours of usable data for analysis.

[5]The difference in the sampling rates for the accelerometers is due to their different use cases, and the importance of sampling minimally so as to preserve battery life: The chest sensor monitors respiration, while the wrist sensors monitor physical activity.

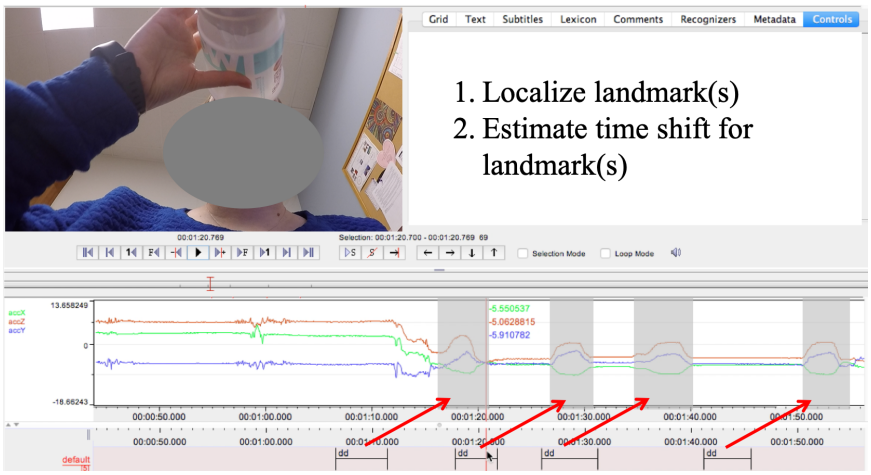


Figure 5.2: Illustration of manual determination of the time shift between a wrist-worn accelerometer and chest-mounted GoPro camera for a drinking gesture, with landmarks defined using the ELAN annotation tool. Once the time-shift is identified, the label for the segmented drinking event can be transferred to the accelerometer signal.

Data Filtering

Because participants can turn on and off the camera at any time, the video clips can be of any length. After removing the clips shorter than 30 seconds, we received a total number of 378 on-body GoPro video clips. We present our dataset screening procedure in a flow diagram in Fig. 5.3. The 378 video clips underwent an initial video quality screening phase in which 31 video clips were excluded due to poor lighting conditions that affected visibility in the recorded video footage, and one additional video clip was removed because the lens was blocked by the wearer's clothing. We filtered data due to sensor quality and hardware error. Since our approach requires a minimum number (20) of 10-second windows with high-quality data (as discussed in Sec. 5.3.2), we excluded 136 clips that did not meet this minimum threshold of data quality. Additionally, when we visualized the sensor data together with the video, we discovered two video clips with erroneous sensor readings, which resulted from a faulty device. We then removed video clips that did not have a landmark or distinguishable human movement (necessary to manually synchronize between the two signals and explained in Sec. 5.2.3). After filtering videos that were not possible to synchronize, we ultimately obtained 163 video clips (45.2 hours) and sensor

data from a total of 21 participants. Each participant contributed, on average, 2.15 hours of "usable" data for analysis.

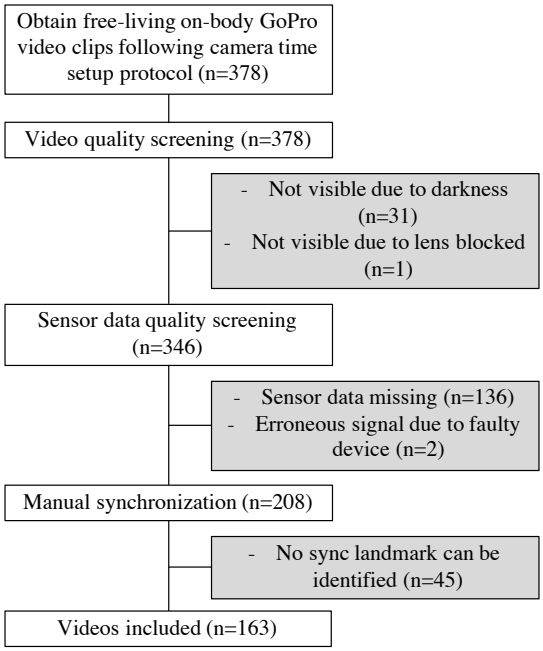


Figure 5.3: Flow diagram of dataset formation.

Ground Truth Annotation Process

Because our data capture process uses separate clocks for the video and accelerometry signals (GoPro clock and study phone clock, respectively), manual alignment of the video and accelerometry signals in each clip was performed in order to establish ground truth for our experiments. In our approach, we chose the accelerometer clock as the reference timeline, and we selected the synchronization offset that shifted the video into alignment.

The manual synchronization process, illustrated in Fig. 5.2, comprises two stages: a landmark video detection phase and an accelerometer alignment phase that aligns the detected video landmarks with one of the two accelerometers (wrist- or chest-worn accelerometer). A video landmark event is defined as a distinguishable human movement (often a transition from inactivity to activity) which is visible in the video and potentially noticeable in either the wrist- or chest-worn accelerometery signals. All accelerometers are

utilized in order to increase the number of available synchronization points. We find that hand movements near the face frequently result in wrist accelerometer-to-video matches, while transitions from sitting to standing or from standing to walking often result in chest accelerometer-to-video matches. After the landmark event has been localized in the video, the annotator utilizes a combination of cues to match sudden changes in either the wrist or chest accelerometer sensor signals to the corresponding hand or body movements in the video. After they are successfully matched, the time offset can then be calculated, and the video time is adjusted to align with the accelerometer based on this offset. The aligned signals are then inspected to determine whether the estimate is sufficiently accurate. If errors remain, the process is repeated for other landmark events, resulting in additional offset measurements. The final offset is produced by combining these measurements. Fig. 5.2 illustrates the process of aligning a drinking event shown in the video frame with the corresponding wrist accelerometer motion using the ELAN [83] annotation tool.

Annotator Agreement

In order to evaluate the consistency of manual labeling, three trained annotators processed 10 video clips and obtained independent estimates of the ground truth offset times. The average difference in their offset times was 346 ms, with 309 ms standard deviation. This result demonstrates an unavoidable error in ground truth acquisition of fine-grained labeling and time synchronization of approximately 346 ms. We also conducted a one-way repeated measures ANOVA (used to determine whether three or more group means are different when the participants are the same in each group). The result illustrates no significant difference among the three annotators (F=0.60, P=0.56). Therefore, we conclude that there is no statistically significant difference between the annotations from different annotators. On average, an annotator spends 25 minutes to synchronize each video (spanning 17 minutes of data) manually. Our work aims to mitigate this problem of time synchronization, thus saving researchers thousands of hours of manual time synchronization.

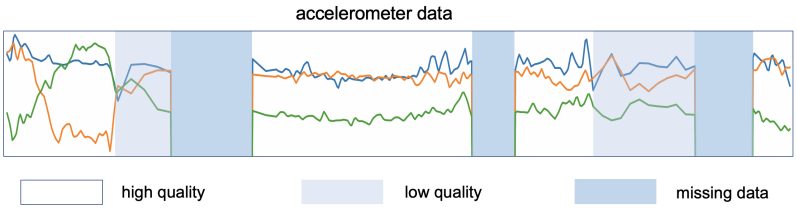


Figure 5.4: A sample of a 3-axis sensor signal. The accelerometer data comprises sections considered either high-quality, low-quality, or missing (no data).

5.3 Methodology

In this section, we introduce our approach to solving the problem of time synchronization between video and accelerometry. We start by presenting the notation used and data preprocessing, and then present our time synchronization algorithm Window Induced Shift Estimation (SyncWISE), and finalize by discussing our evaluation metric.

5.3.1 Notation

We define $\{\mathbf{x_t} \in \mathbb{R}^2\}_{t=1}^{T}$ to be the observed motion acceleration estimated from each frame of video, and $\{\mathbf{y_t} \in \mathbb{R}^3\}_{t=1}^{T}$ to be the observed acceleration from the accelerometer in the chest-worn device. We ensure that the signals are resampled to have the same frequency. Our goal is to estimate the shift $\mathbf{\Delta} \in \mathbb{R}$ between two time series such that $\mathbf{x_t}$ and $\mathbf{y_{t+\Delta}}$ describe the sensed behavior happening at the same time.

5.3.2 Data Preprocessing

Preprocessing has two tasks: 1) identify high-quality windows of accelerometry data to support matching, and 2) extract a motion signal from the video that can be compared to the accelerometry signal.

Accelerometer Data Preprocessing

As illustrated in Fig. 5.4, different portions of accelerometry data can vary significantly in quality due to missing data points, as the result of wireless transmission problems. To

reliably screen for poor-quality data, we analyze the data in 1-second *segments* and define a per-segment reliability metric as the ratio of the number of data points collected in a second divided by the expected sampling rate of 10.66 Hz. We then label each segment of data as high-quality if the reliability is above 75% (*i.e.*, 8 out of the 10.66 samples per second are present). Segments that do not meet this threshold are labeled as low-quality. In order to perform matching, we aggregate high-quality segments into 10-second data *windows*. A viable window will consist of 10 consecutive segments (*i.e.*, a total of 10 seconds) that were labeled high-quality. We sweep across the video clip with a 1-second stride to segment the viable matching windows. In order for a 17-minute video clip to be synchronizable, we require that it contain at least 10 viable windows (each 10 seconds long). Each extracted window is upsampled from 10.66 Hz to 30 Hz to match the camera sampling rate.

Note that our approach uses windows of high-quality accelerometry data as a starting point for matching to windows of video data in our SyncWISE algorithm (discussed in Sec. 5.3.3). This approach assumes that there are no dropped video frames, which is true for the datasets we used in our experiments. Specifically, the GoPro used in S2S-Sync stores frames locally (see Sec. 5.2.2), and the data in CMU-MMAC was collected in a lab setting. It is likely that our approach could be extended to accommodate small numbers of dropped frames, but we make no claims for efficacy in this case. Addressing large numbers of missing frames simultaneously with missing accelerometry is a topic for future work.

Motion Estimation from Videos

In order to compare video and accelerometry, a key operation is to extract an estimate of acceleration from video movement. This is accomplished in two steps. First, an estimate for the velocity at every pixel in every frame, known as optical flow, is computed from each pair of adjacent frames in the video. Motion features have been used in recognizing daily activities from first-person videos [8, 151, 152]. We use a deep-learning based dense optical flow estimation framework called PWC-net [153]. In the resulting 2D vector field,

the vector at each pixel location provides a motion estimate of the pixel in x (horizontal) and y (vertical) directions. The second step extracts a scalar acceleration signal from the sequence of flow fields. To accomplish this, we average the optical flow spatially for each frame and then compute the difference in the average optical flow between the current and previous frames. This approach provides a 2D camera acceleration feature vector for each frame t, denoted as $\mathbf{x_t}$ in Section 5.3.1.

PCA Projection

An important aspect of the video data is that the orientation of the camera can be arbitrary, which consequently affects the orientation of the acceleration vector computed in Sec. 5.3.2. While the participants were instructed to orient the camera towards their head, in practice, we observed many different orientations in the dataset. The orientation can also vary day-to-day as the camera harness is put on and taken off each day. This variation results in coordinate transformations between the sensors that vary both within-subjects across time and between-subjects. While the baseline approach [63] compares signals from a fixed pair of axes to determine the sync points, we need a way to compare axes that vary across time within the dataset.

Principal component analysis (PCA) allows us to map each point of the 2D video data and 3D accelerometery data to a single dimension (1D) that captures the motion along the most dominant axis. To extract the 1D signal that best captures motion in a single axis, we project data from each sensing modality (both the video and accelerometer signal separately) onto their first principal component direction estimated by PCA. The first principal component corresponds to the direction of greatest variation in the data, and this is well-matched to our goal since cross-correlation (used in our proposed algorithm) leverages the variation in the signals. Formally, for video data $\{\mathbf{x_t}\}_{t=1}^{T_{V_k}}$ and accelerometer data $\{\mathbf{y_t}\}_{t=1}^{T_{V_k}}$ collected during video V_k, we have p_{vid1} and p_{acc1} as the first principal components from PCA calculated separately on $\{\mathbf{x_t}\}_{t=1}^{T_{V_k}}$ and $\{\mathbf{y_t}\}_{t=1}^{T_{V_k}}$. We then denote the 1D projected time

series as $\{\mathbf{x_t^{(p1)}} = p_{vid1}^{\mathrm{T}}\mathbf{x_t}\}_{t=1}^{T_{V_k}}$ and $\{\mathbf{y_t^{(p1)}} = p_{acc1}^{\mathrm{T}}\mathbf{y_t}\}_{t=1}^{T_{V_k}}$. We omit the subscription $p1$ for abbreviation throughout the paper.

5.3.3 SyncWISE Algorithm

Our approach to matching noisy video and accelerometry signals captured from mobile devices has two key components. The first is the procedure described in Sec. 5.3.2 and illustrated in Fig. 5.4, which selects high-quality windows of accelerometry data for matching as a way to overcome the noise and missingness in this signal. The second key component uses weighted kernel density estimation (wKDE) to combine noisy estimates of the shift produced from multiple accelerometry-video window pairs to obtain an accurate estimate. It utilizes the cross-correlation (CC) response from each window pair to obtain a weighted Gaussian kernel and combines these kernels to estimate the offset.

The process of offset estimation is illustrated schematically in Fig. 5.5. The top of the figure shows the signals from a video clip and corresponding accelerometry clip separated by a ground truth offset of 1.5s. Given a window of accelerometry data, we search for the offset by shifting the video window by different amounts, to examine corresponding segments in the video. This is illustrated at the top of the figure. We show three different accelerometry windows in blue, cyan, and green. Each window will be shifted multiple times to search for potential matches. One of the corresponding window locations in the video signal is shown as a solid outline, while the other locations are shown with dotted outlines. Given each pair of windows, CC is used to produce an estimate of the shift from that pair. This step is illustrated in the lower part of Fig. 5.5 for each of the three window pairs shown with solid outlines. After the peak in the CC function is detected, a Gaussian kernel is fitted to the data. Once all pairs have been correlated, a probability density function (PDF) for the global offset between the signals is constructed via a weighted sum of the Gaussian kernels (i.e., a wKDE for the offset PDF). Effectively, each window pair is casting a weighted vote for the offset. A single estimate for the offset is obtained by

detecting the peak in the PDF, resulting in an estimate of 1.4s with an error of 100ms in this schematic example. The confidence score for the final estimate is obtained by fitting a Gaussian to the PDF to obtain the variance (24 in this example). Pseudocode for the

method is provided in Algorithm 1. We now describe each step in detail.

Algorithm 1: SyncWISE: Synchronization based on Window Induced Shift Estimation

Input : video and accelerometer data clip with asynchronous clocks

Output: time shift δ of video and accelerometer data and confidence C

Obtain optical flow of video;

Calculate 1-component PCA of optical flow as $x_1, x_2, \ldots, x_k$ and 3-axis accelerometer data as $y_1, y_2, \ldots, y_k$;

Data screening on accelerometer data to obtain T_w-second windows of high-quality data, and resample to the same sampling rate as the optical flow signal;

Initialize N_s, T_{max}, T_w;

$N \leftarrow$ number of windows;

for *i-th window* $(s_i, s_i + T_w)$ **do**

  for $j = 1, 2, \ldots, N_s$ **do**

    $o_j \leftarrow random(-T_{max}, T_{max})$;

    $\vec{x} \leftarrow x_{s_i+o_j}, x_{s_i+1+o_j}, \ldots, x_{s_i+T_w+o_j}$;

    $\vec{y} \leftarrow y_{s_i}, y_{s_i+1}, \ldots, y_{s_i+T_w}$;

    $cc(\tau) \leftarrow$ cross-correlation of $\vec{x}$ and $\vec{y}$;

    $\delta_{ij} \leftarrow \arg\max_\tau \left| cc_{i,j}(\tau) - \underset{t\in[-T_w,T_w]}{\text{median}} \{cc_{i,j}(t)\} \right| + o_j^i$;

    $conf_{ij} \leftarrow \frac{\max_\tau \left| cc_{i,j}(\tau) - \underset{t\in[-T_w,T_w]}{\text{median}} \{cc_{i,j}(t)\} \right|}{\underset{t\in[-T_w,T_w]}{\text{std}} \{cc_{i,j}(\tau)\}}$;

  end

end

$f(t) \leftarrow 0$;

for $i = 1, 2, \ldots, N$ **do**

  for $j = 1, 2, \ldots, N_s$ **do**

    $f(t) \leftarrow f(t) + conf_{ij} \cdot K(\delta_{ij}, t), K() = Gaussian()$;

  end

end

$f(t) \leftarrow \frac{1}{\sum_i \sum_j conf_{ij}} f(t)$;

$\delta \leftarrow \arg\max_t f(t)$;

Fit Gaussian curve $\mathcal{N}(\mu, \sigma)$ to $f(t)$;

$C \leftarrow \frac{Const}{\hat{\sigma} \cdot var(\hat{\mu})}$

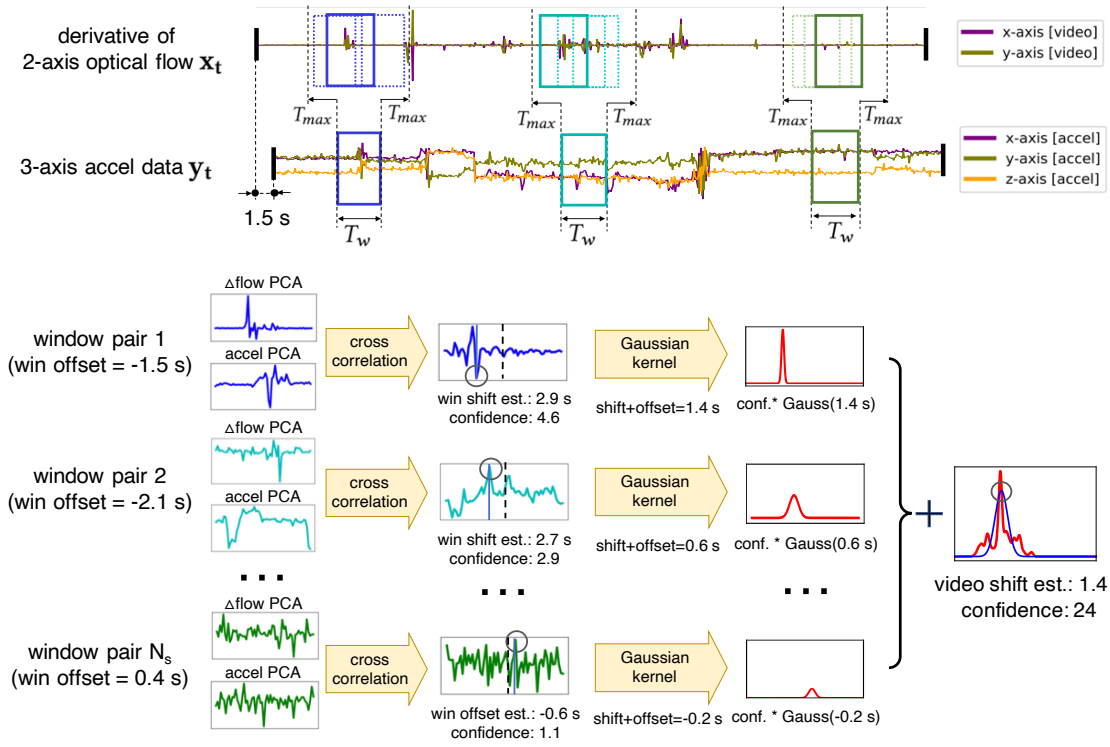


Figure 5.5: SyncWISE algorithm overview: For each window that contains high-quality accelerometry data, we search for the offset by shifting the video window by different amounts to examine corresponding segments in the video (shown as blue, cyan, and green boxes). Given each pair of windows, cross-correlation (CC) is used to produce an estimate of the shift from that pair. A probability density function for the global offset between the signals is constructed via aggregating estimates from all window pairs using weighted kernel density estimation as illustrated in the lower part of the figure.

Window Pair Sampling

Our starting point is a window w_i of accelerometry data of length T_w with samples denoted as $\mathbf{y^i} = [y_{s_i}, y_{s_i+1}, \ldots, y_{s_i+T_w}]$, where s_i is the time index of the start of window i. We generate N_s different offsets o^i_j for window i, where $j = 1, \ldots, N_s$ and $|o^i_j| <= T_{max}$. For each offset j, we obtain corresponding video samples denoted as $\mathbf{x^{i,j}} = [x_{s_i+o^i_j}, x_{s_i+o^i_j+1}, \ldots, x_{s_i+o^i_j+T_w}]$. Each o^i_j defines a window pair (see Fig. 5.5) for matching via CC in Sec. 5.3.3.

This approach has three design parameters: T_w, which controls the window width; T_{max}, which controls the search range; and N_s, which controls the number of offsets. These

parameters are illustrated in Fig. 5.5. The consequences for these parameter settings are discussed in Sec. 5.3.3, and their optimization is discussed in Sec. 5.4.4. Note that there are many possible ways to generate the N_s offsets. The most straightforward approach would be to uniformly sample the search range in fixed steps. Alternatively, if a prior estimate for the shift is available (for example, from knowledge of the capture setup or manual inspection), then sampling from a prior distribution could focus the window comparisons on the more likely offsets (as in importance sampling). In our experiments, we sampled offsets at random from a uniform distribution over T_{max}. We confirmed experimentally that this approach was indistinguishable from covering the search range in fixed steps.

Window Pair Matching

Given accelerometry-video window pairs $\mathbf{y^i}$ and $\mathbf{x^{i,j}}$, as defined in Sec. 5.3.3, we match them by calculating the CC [63, 80] to obtain an estimate $\delta_{i,j}$ for the shift between the clips. Specifically, we calculate the CC function:

$$cc_{i,j}(\tau) = \sum_{n=1}^{T_w} y_n^i * x_{n+\tau}^{i,j}, \quad \tau \in [-T_w, T_w], \tag{5.1}$$

using an efficient fast Fourier transform (FFT)-based approach. The optimal shift for this pair of accelerometery and video data is then estimated by choosing the τ that maximizes the absolute value of the CC function centered by its median (to normalize the values) and shifted by o_j^i (its pregenerated offset):

$$\delta_{i,j} = \arg\max_{\tau} \left| cc_{i,j}(\tau) - \underset{t\in[-T_w,T_w]}{\text{median}} \{cc_{i,j}(t)\} \right| + o_j^i. \tag{5.2}$$

Note that each $\delta_{i,j}$ represents an estimated shift obtained from accelerometry window w_i and offset o_j^i. We then obtain a confidence score for the estimated shift as follows:

$$conf_{i,j} = \frac{\max_\tau \left| cc_{i,j}(\tau) - \underset{t\in[-T_w,T_w]}{\text{median}} \{cc_{i,j}(t)\} \right|}{\underset{t\in[-T_w,T_w]}{\text{std}} \{cc_{i,j}(\tau)\}}. \tag{5.3}$$

The confidence score captures the fact that not all window pairs are equally informative regarding the true offset. Intuitively, higher and sharper peaks are associated with more reliable estimates.

Synchronization Offset Estimation

The procedures described in Secs. 5.3.3 and 5.3.3 are repeated M times for windows of accelerometry data selected from the input clip. This results in a set of MN_s shift estimates and associated confidence values $\{\delta_{i,j}, conf_{i,j}\}$. These inputs are used to obtain a weighted kernel density estimate (wKDE) for the distribution of possible offset times between the two clips. This estimate is constructed via a confidence-weighted voting approach as follows:

$$f(t) = \frac{1}{\sum_{i=1}^{M}\sum_{j=1}^{N_s} conf_{i,j}} \sum_{i=1}^{M}\sum_{j=1}^{N_s} conf_{i,j} \cdot K(t|\delta_{i,j},\sigma), \tag{5.4}$$

where $K(t|\delta,\sigma)$ is a Gaussian kernel function with mean δ and standard deviation σ. The function $f(t)$ is a pdf over the range of offsets for the input clips, which is constructed by summing weighted Gaussian kernels from the M windows and N_s shifts. The final estimated video shift Δ is attained where $f(t)$ is maximum, as follows:

$$\Delta = \arg\max_t f(t). \tag{5.5}$$

It can be valuable to have a confidence score associated with the estimated offset Δ.

For example, given a pair of clips to be matched, the confidence score could be used to determine whether the number of window samples M is sufficient. We estimate the confidence in the offset estimate by using nonlinear least squares [154] to fit a Gaussian curve $g(t) = \mathcal{N}(\mu, \sigma)$ to $f(t)$, and using the variance $\hat{\sigma}$ and the variance of mean $var(\hat{\mu})$ of the Gaussian to inform the confidence. The intuition here is that, if the variance of the Gaussian is high, then the confidence in the delta offset is lower, and if the variance of the estimated mean is high, then the confidence is also lower. The estimated variance $\hat{\sigma}$ and the variance of the estimated mean $var(\hat{\mu})$ predicts how well $f(t)$ fits a Gaussian distribution. We define the confidence score as follows:

$$C = \frac{Const}{\hat{\sigma} \cdot var(\hat{\mu})}. \tag{5.6}$$

In this work, $Const$ is empirically set to be 200,000. Appendix 5.4.3 gives some examples of this process.

Discussion of the Impact of Parameter Choices

The method depends on the choice of window size T_w, maximum offset T_{max}, and number of window pairs (shifts) N_s. Only windows sampled near the ground truth offset can contribute positive votes to the response curve $f(t)$. The percentage of positive votes is at most T_w/T_{max}. Therefore, larger T_w and smaller T_{max} values are preferred. However, the effect of T_w also depends on data quality, and when T_w is too large, it may be difficult to find a sufficient number of windows with high-quality samples. Similarly, T_{max} should be larger than the maximum expected true offset. N_s should be sufficiently large to cover the search range with candidate sync points, but the only upper bound on its value is computational resources. In general, the optimal choice of these parameters will be dataset-dependent.

5.4 Experiments and Results

We performed extensive experimentation in order to evaluate the proposed SyncWISE algorithm. We evaluated four different synchronization methods on two datasets: our novel S2S-Sync dataset and the CMU-MMAC kitchen activities dataset [77]. The algorithm variations are detailed in Sec. 5.4.1, followed by a discussion of error metrics in Sec. 5.4.2. Experiments on S2S-Sync and CMU-MMAC are detailed in Secs. 5.4.3 and 5.4.5, respectively.

5.4.1 Algorithm Variations

We identify four different synchronization methods that vary according to their treatment of the partial observability and coordinate registration problems. Our baseline method is the approach in [63], which performs a single global CC. In the S2S-Sync dataset, we fill any missing accelerometry data by carrying over the last observation. We report results for two variants of this core approach.[6] The first is *Baseline-xx*, in which we select the x component of both the accelerometry and camera acceleration features, defined in Sec. 5.3.2. This choice was motivated by the orientation of the sensors (see Fig. 5.1(b)), as the x-axis corresponds to side-to-side motion and is most likely to result in a strong motion signal during movement in real-world settings.[7] The second variant is *Baseline-PCA*, for which we use the PCA-based approach described in Sec. 5.3.2 to determine the 1D signals used for CC. The SyncWISE family of methods uses the paired window matching approach with wKDE described in Sec. 5.3.3. The variant *SyncWISE-xx* uses the x-axis coordinate choice described above, while *SyncWISE* incorporates the PCA representation. *SyncWISE* is the best-performing variant across both datasets and it establishes the new state-of-the-art for this problem. Note that in order to ensure a fair comparison, before applying CC to

[6] We used the open source reference implementation provided by the authors and incorporated it into our codebase so we could easily implement preprocessing and other steps. Our code and data are available at `https://github.com/HAbitsLab/SyncWISE`.

[7] We verified experimentally that selection of the y-axis direction results in worse performance.

Baseline, we drop all low-quality windows from each clip and concatenate the high-quality windows together in both video and accelerometry, thereby ensuring that both methods see the same signals as input.

5.4.2 Evaluation Metric

We use two measures to evaluate algorithm performance: 1) the average of the absolute value of the synchronization error, E_{avg}, and 2) the percentage of clips that are synchronized to an offset error of less than n ms, PV-n. The choice of n connects to the question of how accurate temporal synchronization needs to be in order to be useful in practice. In general, the answer to this question will be application-dependent. For example, a smoking puff can be as short as 500ms, but smoking sessions last 5-7 minutes [155], and teeth brushing lasts 2 minutes [76]. We note that the average annotator disagreement in S2S-Sync was 346ms, and prior works such as [82] used 300ms as the accuracy target. Based on these considerations, we chose to report *PV-300* and *PV-700* as the accuracy measures in our experiments. The PV-n measure is complementary to E_{ave}, which can be sensitive to outliers if a few difficult videos produce very large synchronization errors.

5.4.3 Experimental Results for S2S-Sync Dataset

In the following sections, we conduct two experiments to validate the performance of our method and describe the parameter settings. The first experiment compares the performance of Baseline and SyncWISE using a simulated dataset with randomly generated offsets to provide a large-scale evaluation. Example response curves are provided. The second experiment demonstrates the method's ability to synchronize the original clips in S2S-Sync using an iterative extension of our basic algorithm. In addition, Sec. 5.4.4 describes a comprehensive sensitivity analysis on a held-out dataset, which illustrates the impact of parameter changes on performance. All findings are discussed in Sec. 5.4.3.

Parameter Specification

As described in Sec. 5.3.3, the choice of parameters for the method is dataset-dependent in general. We now detail the parameters used in all experiments in this section. The relatively high amount of missing data in the accelerometry signal motivated the choice of $T_w = 10$s. We performed a sensitivity analysis, detailed in Sec. 5.4.4, to characterize the effect of N_s and T_{max} on the performance. We selected $N_s = 20$ and $T_{max} = 5$s. We used $\sigma = 500$ for the wKDE in Equation 5.4, and set the search bounds for μ and σ in obtaining confidences for Equation 5.6 to be $[-20,000ms, 20,000ms]$ and $[0, Inf]$, respectively.

SyncWISE Compared to Baseline

We applied a random 20/80 split of the 163 video clips, where 20% (33 videos) are used to optimize the parameters of our algorithm, and 80% (130 videos) are used to test our algorithm and produce the final results. Using the ground truth synchronization offsets for the 130 test video clips (see Sec. 5.2.3), we generated a synthetic testing dataset as follows: random offsets in the range [-3 sec, 3 sec] were sampled 30 times for each clip and used to shift the synchronization, resulting in 3,900 test clips. These synthesized test clips were used to assess the performance of our algorithm.[8] The experimental results are summarized in Table 5.1. We can see that E_{avg} is two orders of magnitude smaller for SyncWISE compared to Baseline. This difference is not surprising, as Baseline does not aggregate votes based on a quality measure and is therefore susceptible to the prevalent signal noise. Similarly, the PV-300 and PV-700 measures are higher for SyncWISE, although the difference is not as large, suggesting that a smaller set of difficult clips may explain the high E_{avg} result for Baseline. We see that the use of PCA in SyncWISE provides a modest benefit relative to SyncWISE-xx, particularly for PV-300, but Baseline-PCA performs worse than

[8]Since SyncWISE searches in the space of offsets and evaluates a discrete set of windows, it will produce different outputs for different relative shifts. In contrast, the Baseline approach uses a single global cross-correlation, and the shift-invariance property of the cross-correlation function means that the results will be the same for all shifts. Therefore, the Baseline method was run once for each clip.

Table 5.1: Results on S2S-Sync Dataset for baseline-xx, baseline-PCA, SyncWISE-xx and SyncWISE (T_w=10s, T_{max}=5s, N_s=20) with random shift. The SyncWISE results are averaged over 30 runs. In each run, we generate a random number from [-3 sec, 3 sec] as the ground truth shift between videos and accelerometer data. Baseline results are based on a single run because it is not affected by different input shift and no randomization is involved in this algorithm.

Method	# Videos	Ave #Win Pairs	Ave Error (ms)	PV-300 (%)	PV-700 (%)
Baseline-xx [63]	130	1	29690.79	50.77	82.31
Baseline-PCA	130	1	51124.77	47.69	70.77
SyncWISE-xx	130	1403.45	447.01	62.36	**89.72**
SyncWISE	130	1403.45	**416.30**	**73.38**	88.72

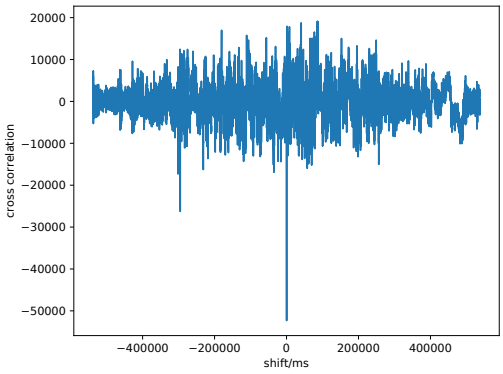
Baseline-xx. We hypothesize that this is because the PCA step primarily benefits the quality of the confidence estimate, which is not used in Baseline. We include some examples of response curves for both methods in the following sections.

Examples of Response Curves

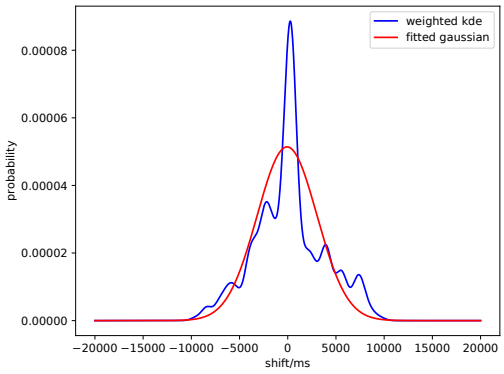
Figure 5.6 shows example response curves of both methods for two sessions using the wearable camera and chest accelerometer. Baseline method fails in the second session because of the irregular sampling rate and sparse signal segments in our dataset.

Examples of Confidence Scores

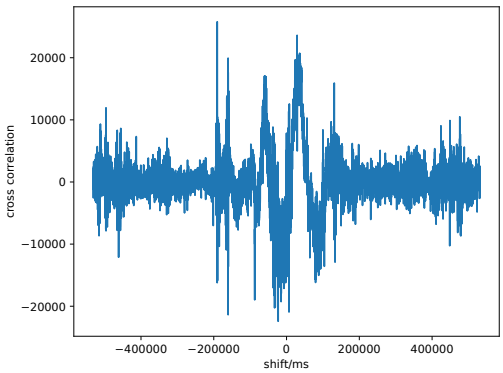
Figure 5.7 shows some examples of voted shift curves $f(t)$ and their corresponding confidence score. Confidence in (a) and (b) is low due to large variance and spurious peaks. Figure(c) shows an example of high confidence.



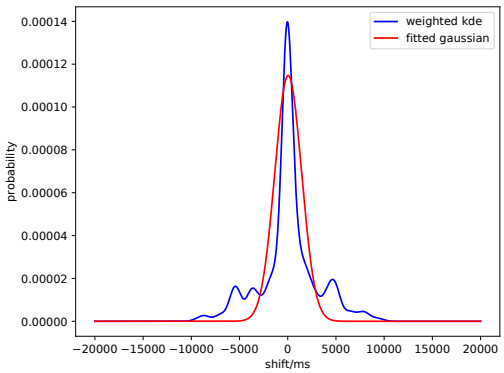
(a) Baseline-xx estimation on session 1 is -267 ms



(b) SyncWISE estimation on session 1 is 261 ms



(c) Baseline-xx estimation on session 2 is 190426 ms

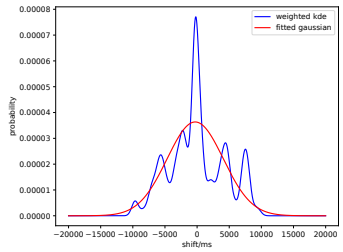


(d) SyncWISE estimation on session 2 is -34 ms

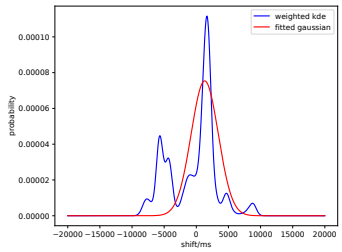
Figure 5.6: Example response curves of both methods for two sessions using a wearable camera and chest accelerometer. Ground truth for both sessions is 0s. (a) and (c) show the CC function of baseline method. (b) and (d) show $f(t)$ of SyncWISE for the same sessions. In (b), 480 windows are sampled, and the confidence score of the estimation is 0.89. In (d), 2140 windows are sampled, and the confidence score of the estimation is 10.11.

Synchronizing With the Original Offsets

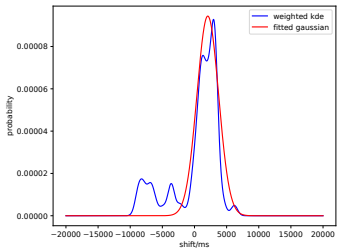
The simulation experiment in Sec. 5.4.3 facilitated a large-scale evaluation over a 6-second range of offsets. However, the ground truth offsets for the S2S-Sync clips are substantially larger and have a complex distribution, with an average offset of 21s, max offset of 180s, and min offset of 387ms. The direct application of SyncWISE to these clips is unsuccessful, as the search range is too large for accurate registration given the noisy and complex properties of these signals.



(a) Ground truth shift = -325 ms



(b) Ground truth shift = 1827 ms

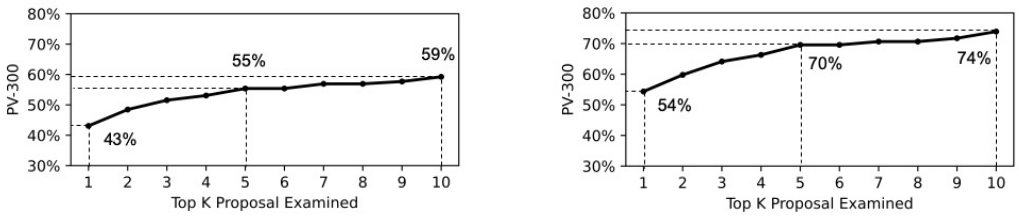


(c) Ground truth shift = 2755 ms

Figure 5.7: Examples of video shift and confidence estimation. In (a), 500 windows are sampled, SyncWISE estimation is -206 ms, and the confidence score of the estimation is 0.19. In (b), 260 windows are sampled, SyncWISE estimation is 1651 ms, and the confidence score of the estimation is 1.13. In (c), 460 windows are sampled, SyncWISE estimation is 2958 ms, and the confidence score of the estimation is 9.37. Confidence in (a) and (b) is low due to large variance and spurious votings. Figure(c) shows an example of high confidence. The fitted Gaussian successfully finds the true peak and ignores the false peaks.

We therefore developed an interactive version of the SyncWISE method which can extend the search range arbitrarily, with the downside of requiring human intervention for verification. We draw an analogy to image searches to explain this issue. When the query is easy, the first returned result will have high confidence and can be accepted immediately. This is the case for synchronization with a 6-second offset range. However, when the query is challenging, confidence must be used to rank the results, and manual inspection is needed to verify the correct match. This is because the confidence estimates are not sufficiently accurate over an extremely large search range. We now describe the *Extended SyncWISE* approach as follows: We define a step size V_{step} of 3 s and a maximum shift V_{max} of 3 min. These are chosen to search over the range of +/- 3 min, which is sufficient to cover all of the offsets in the S2S-Sync clips. We generate a set of shifted clips by starting with $-V_{max}$ and incrementing to $+V_{max}$ in steps of V_{step}, resulting in our case in 120 shifted clips. Each shifted clip is synchronized using SyncWISE, and the offsets from these clips are ranked according to the confidences computed via Equation 5.6. We then ask the user to examine the proposed sync points in ranked order to identify a desired level of accuracy.

Since we already have the ground truth sync points for the purpose of this experiment,



(a) (b)

Figure 5.8: Result of the Extended SyncWISE method on the S2S-Sync dataset with the original offsets (unsynchronized clips). Ranked proposals for sync points are examined to retrieve the correct sync. The plot shows the relationship between the number of top-ranked proposals examined and the resulting PV-300 measure (a) on 130 videos in test dataset, and (b) on 92 videos after removing problematic videos.

the user role can be automated. The results are depicted in Fig. 5.8. The x-axis shows the number K of top-K ranked proposal (candidate) sync points that must be examined, and the y-axis gives the percentage of clips correctly synchronized according to the PV-300 measure if the best proposal within the top K is selected. In the 130 videos in the test dataset, several had less than ideal recording conditions: five videos were recorded under dim lighting conditions; one video was blurred by water droplets on the lens; nine videos had the edge of the lens covered by the GoPro case; and 23 exhibited little to no salient body movement. In Fig. 5.8, we report on both the group of 130 videos and of 92 videos (after removing these problematic videos). After removal of problematic videos, we can observe that simply choosing the top-ranked sync proposal (as in the standard SyncWISE approach), results in a PV-300 of approximately 54%. However, examining the top five proposals permits an improvement to approximately 70%, while examining the top 10 proposals achieves a PV-300 of 74%. While this result falls short of a fully automated approach, our method has the benefit of broad applicability and may still save significant human effort in synchronizing video clips under difficult and real-world recording conditions.

Discussion

Our results demonstrate the benefits of our SyncWISE approach over the Baseline global CC method. This result is reasonable as the baseline approach assumes that all parts of the video and accelerometry signals contain information that is relevant to the offset and combines them into a single global estimate. This assumption is unlikely to be true in our case due to partial observability. Our method attempts to automatically find the most salient windows with high correlation to act as "marker gestures," similar to those defined manually in synchronizing data from a controlled in-lab study in [80, 82]. The problem of sychronizing clips over very long offsets in the presence of sensor noise remains open, but the Extended SyncWISE approach can provide a stop-gap interactive method.

Use of SyncWISE requires the user to set the system parameters, and we included a sensitivity analysis in our experiments which leveraged the known ground truth. In general, it will be difficult to specify the correct system parameters in advance for all sync problems. In practice, we believe an iterative approach will be needed, in which an initial set of parameters is refined through repeated synchronization and analysis of a small set of clips. Once effective parameters are found, then an entire dataset can be synchronized automatically, or interactively using Extended SyncWISE.

5.4.4 Parameter Optimization on S2S-Sync Dataset

We perform sensitivity analysis (parameter optimization) using 20% (33 of the 163) of the videos. We set the Gaussian kernel σ to 500. We also optimize the following three main parameters (T_w, T_{max}, and N_s).

Optimization for Window Size T_w

As described in Sec. 5.3.2, since we eliminate the seconds with low-quality data, if T_w is too large, then it may disqualify several windows due to the 80% sampling rate quality threshold. When we set T_w as 10 s, after the data screening step, the 33 videos each have

Table 5.2: Parameter T_{max} search (T_w=10 s, N_s=20) for S2S-Sync dataset

Input Shift (s)	T_{max} (s)	Ave Error (ms)	PV-700	PV-300	Ave Conf.
0	2	268	0.94	0.79	892
0	4	**262**	0.94	0.79	96
0	6	284	0.94	0.76	11
0	8	364	0.94	0.76	1
0	10	282	0.94	0.76	0
2	2	447	0.91	0.79	106
2	4	307	0.91	0.79	18
2	6	**264**	0.91	0.76	5
2	8	597	0.88	0.73	1
2	10	753	0.88	0.67	0
4	2	947	0.79	0.67	1
4	4	923	0.82	0.7	1
4	6	786	0.85	0.73	0
4	8	814	0.88	0.73	0
4	10	**716**	0.88	0.76	0

on average 82 qualified high-quality windows. When we adjust T_w to 20 s, 29 videos no longer have a sufficient number of qualified windows. After further investigation we set the window size at 10 s to ensure that a sufficient number of videos can be processed and that we can address low-quality video-sensor pairs.

Optimization for Max Random Offset T_{max}

We tested a range of values for T_{max} from 2 s to 10 s with input data of different shifts from 0 to 4 s. As shown in Table 5.2, when the input shift is 2 and T_{max} is set to 6 s, the average error is 264 ms. The average error increases to 753 ms with increasing T_{max}. When the input shift is 0 s, the average errors are 268, 262, and 284 ms when T_{max} is 2 s, 4 s, and 6 s, respectively, showing no significant difference. With a 4 s input shift, the minimum average error occurs when T_{max} is set to 10 s. When we apply SyncWISE in a real-world setting, due to a lack of the prior knowledge of the input shift, we select T_{max} to be 5 s (average between 4 and 6), as long as the target offset range is less than 4 s.

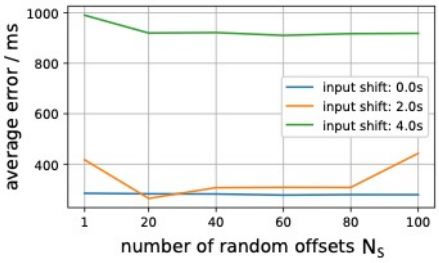


Figure 5.9: Average error with different number of random offsets for each window (T_w=10 s, T_{max}=3 s).

Optimization for Number of Random Offsets N_s

We then fixed T_w and T_{max} and adjusted N_s. As shown in Fig. 5.9, when we adjusted N_s in the range from 20 to 80, with input data of different shifts of 0 and 2 s, the average error did not significantly change, suggesting that 20 offsets within a window is sufficient. Fig. 5.9 shows the average error as a function of N_s. We selected N_s to be 20, which provides the least average error.

5.4.5 Experimental Results for CMU-MMAC Dataset

We now describe our experiments on CMU-MMAC, a well-known multimodal activity dataset [77]. The data of different modalities used in this experiment and parameter settings are described in Sec. 5.4.5. The experiment in Sec. 5.4.5 compares the performance of Baseline and SyncWISE using the ground truth synchronization provided with the dataset. Example response curves are provided. In addition, we provide additional results on all IMU positions and different parameter combinations and compare random and regular spacing window sampling methods. The findings are discussed in Sec. 5.4.5.

CMU-MMAC Dataset

In this dataset, a wearable camera is attached to a head lamp whose bulb has been removed. This rig is worn around the subject's head, and five accelerometer sensors are placed on the subject's back, legs, and arms. Data is recorded by a laptop through wired connections,

Table 5.3: Final result on the CMU-MMAC Dataset for synchronization between wearable camera and right arm accelerometor using SyncWISE, Baseline-PCA and Baseline-xx (with T_w=60 s, T_{max}=60 s, N_s=10).

Method	# Videos	Ave #Win Pairs	Ave Error (ms)	PV300	PV700
Baseline-xx [63]	126	1	26371.69	30.16	43.65
Baseline-PCA	126	1	21983.33	40.48	50.0
SyncWISE	126	2865.32	**14358.99**	**49.21**	**63.49**

and synchronization between the signals is achieved through network time sync protocols. The ground truth offsets to achieve synchronization are within the range of [1.8 s, 178.0 s], with a mean of 26.4 s. The videos are collected at 30 fps and accelerometry data at 125 Hz. There are 126 sessions from 30 subjects with valid video and IMU data. In this experiment, we use the unsychronized video and accelerometry as the input to our method.

Parameter Specification

We interpolate the accelerometry data to match the video sampling rate. We choose random offsets o_j^i over three ranges, from 0 to 60, 90, or 120, to generate an initial offset for synchronizing and then apply windowed cross-correlation. Note that the maximum temporal offset we can recover with our method is the sum of the maximum random offset amount T_{max} and the window size T_w. In this dataset, we set $N_s = 10$. We select a Gaussian kernel $K(\delta, t)$ with $\sigma = 3000ms$. We set the search bounds for estimating μ and σ of $g(t)$ to be $[-600s, 600s]$ and $[0, Inf]$, respectively.

SyncWISE Compared to Baseline

Table 5.3 shows the results from the baseline method and the SyncWISE algorithm. The input clips combine the wearable camera signal with the accelerometer data coming from the right arm. We include the results for pairing video with the accelerometers from the

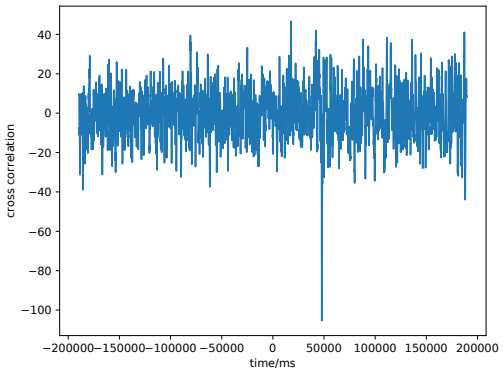
Table 5.4: Baseline results on CMU-MMAC

IMU Position	Baseline-xx			Baseline-PCA		
	Average Error	PV-300	PV-700	Average Error	PV-300	PV-700
Left Arm	32687.30	12.70	16.67	29832.01	20.63	30.16
Right Arm	26371.69	30.16	43.65	21983.33	40.48	50.00
Left Leg	50740.48	2.38	7.14	46670.90	7.94	13.49
Right Leg	52933.07	1.59	4.76	49400.26	3.97	8.73
Back	53221.43	3.17	5.56	29112.96	23.81	30.95

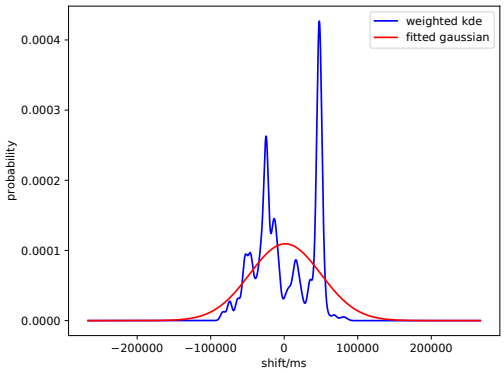
other locations, along with different parameter combinations and examples of the response curves from both methods, in the following sections. This dataset is challenging because none of the accelerometery sensors are co-located with the camera. The frequent relative movement between the sensor and camera creates challenges for the baseline method. From Table 5.4, we can see that PCA improves the performance of the baseline, presumably by identifying the corresponding axes across modalities in the face of relative movement of the sensors. Our window-weighted kernel density estimation approach further improves performance consistently across all of the different accelerometer positions. This demonstrates the ability of our method to automatically focus on the "signal intensive" temporal windows and ignore those of little synchronization utility.

Examples of Response Curves in CMU-MMAC Dataset

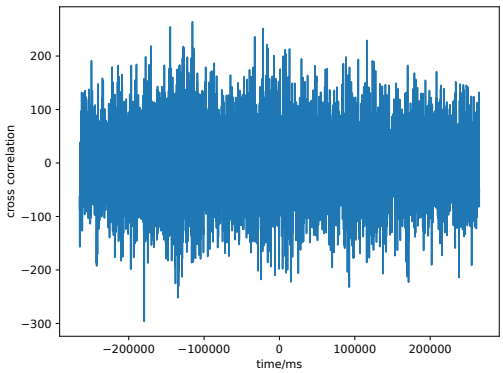
Figure 5.10 shows example response curves of both methods for two sessions using the wearable camera and right-arm accelerometer in the CMU-MMAC dataset. Both methods work well in the first session. The baseline method failed in the second session because this session contains less hand movement with high frequency and high magnitude across the video.



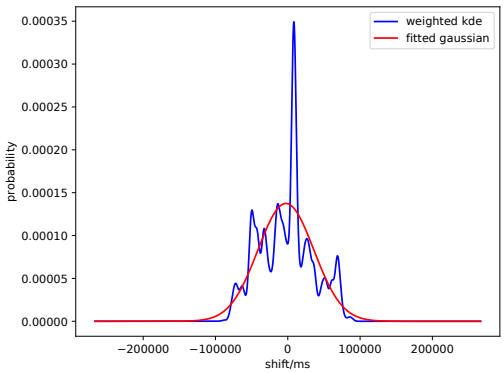
(a) Baseline-PCA estimation on session 1 is 47.800 s



(b) SyncWISE-60/60 estimation on session 1 is 47.733 s



(c) Baseline-PCA estimation on session 2 is -179.433 s



(d) SyncWISE-60/60 estimation on session 2 is 8.400 s

Figure 5.10: Example response curves of both methods for two sessions using wearable camera and right-arm accelerometer. (a) & (b) Ground truth 47.700 s, (c) & (d) Ground truth 8.000 s. In (b), 3190 windows are sampled, and the confidence score of the estimation is 0.05. In (d), 4710 windows are sampled, and the confidence score of the estimation is 0.39.

Results for all IMU Positions and Different Parameter Combinations

As shown in Table 5.5, among all window size/maximum random offset configurations, SyncWise-60/60 achieves the best performance. This result supports our analysis in Sec. 5.3.3 that large window sizes and small maximum random offsets are preferred in this algorithm.

Comparing results across different IMU positions, we find that right-arm acceleration is most informative in synchronizing with the camera using both approaches, followed by the back sensor, with the left and right leg data being the least informative modalities.

Table 5.5: SyncWISE results on CMU-MMAC

IMU Position	SyncWise-30/60			SyncWise-30/90			SyncWise-30/120		
	Ave Error	PV-300	PV-700	Ave Error	PV-300	PV-700	Ave Error	PV-300	PV-700
Left Arm	19539.15	22.22	43.65	20157.14	24.60	43.65	26284.92	19.84	33.33
Right Arm	15902.12	43.65	**65.87**	17626.72	39.68	60.32	19237.83	34.13	53.17
Left Leg	19738.36	22.22	42.06	20122.22	12.70	39.68	23418.52	11.90	29.37
Right Leg	16624.07	20.63	42.06	20339.68	12.70	36.50	21445.50	13.49	30.95
Back	16755.03	43.65	59.52	18342.59	38.88	56.35	18522.22	38.89	53.97
IMU Position	SyncWise-60/60			SyncWise-60/90			SyncWise-60/120		
	Ave Error	PV-300	PV-700	Ave Error	PV-300	PV-700	Ave Error	PV-300	PV-700
Left Arm	16808.99	25.40	43.65	17866.67	24.60	44.44	18337.30	22.22	40.48
Right Arm	14358.99	**49.21**	63.49	13194.71	46.83	62.70	12669.58	42.06	61.11
Left Leg	20927.25	10.32	28.57	19902.38	11.90	31.75	25044.44	9.52	26.98
Right Leg	20777.78	13.49	32.54	22119.58	11.11	27.78	22855.29	9.52	25.40
Back	15759.26	39.68	53.17	16034.39	35.71	52.38	15425.13	34.13	53.17

Table 5.6: Experiment on window sampling methods

	Evenly Spaced Sampling			Random Sampling		
IMU position	Ave error	PV-300	PV-700	Ave error	PV-300	PV-700
Left Arm	25526.67	0	20	29263.33	0	30
Right Arm	38446.67	50	50	38433.33	50	50
Left Leg	35470.00	20	20	35486.67	20	30
Right Leg	46040.00	10	20	39883.33	10	20
Back	29760.00	40	50	29776.67	40	50

This result is consistent with our findings. In this kitchen dataset, the subjects perform many actions using their right hand, making the camera vibrate accordingly. In contrast, leg motion exerts less impact on the camera located on the subject's head. This finding is interesting because we expected the IMU on the subject's back to have the least relative motion to the camera and, consequently, to yield the best synchronization performance. However, the results show that variability in the data, representing frequent motion, is essential for successful synchronization, even when the camera and accelerometer are not co-located.

Comparing Window Sampling Methods

We randomly selected 10 videos from the CMU-MMAC dataset to validate our choice of random offset instead of evenly placed offsets. Table 5.6 shows results using evenly spaced sampling and random sampling. The window size and maximum search range are both set at 60 s. $N_s = 20$ window pairs are sampled for each sliding window in accelerometer data. The two sampling methods do not yield significant performance difference.

Discussion

In the case of the CMU-MMAC dataset, where data is collected with a fixed sampling rate and no data frames are dropped during capture, we are able to select a large window size T_w without worrying about problems arising due to low data quality, and use a reasonable empirically set value for T_{max}. Nonetheless, the performance of all methods on this dataset are lower than that of the S2S-Sync dataset, which suggests that the large relative motion between the camera and the accelerometers, as a result of their positions on the body, may be amplifying the challenge to performing the sync task. The experiments further demonstrate that parameter choices are dataset-dependent. Even for the same dataset, the best performance for accelerometers located at different positions is achieved by different parameter choices.

5.5 Discussion, Conclusion and Future Work

The ability to accurately synchronize multiple data streams is a longstanding problem with increasing utility for researchers who seek to develop and validate computational models to detect daily human behaviors from multimodal wearable sensor suites in the wild. Specifically, as visual confirmation is a widely used method to label target events, an accurate, feasible, less burdensome means to synchronize video with adjacent sensor streams has become urgently needed.

In the current study, we demonstrated the feasibility and effectiveness of a novel approach (SyncWISE) to synchronize data from a wearable video camera with data from a wearable accelerometer. SyncWISE addresses the problems of partial observability (where sensors do not capture the same events) and coordinate transformation (sensor axes are not spatially-aligned over time) that characterize challenging real-world synchronization tasks involving data collected from wearable sensors. We evaluate SyncWISE on two datasets: a novel smoking cessation dataset S2S-Sync and the CMU-MMAC dataset [77]. We demonstrate state-of-the-art performance relative to a recent baseline method.

Although our work focuses on resolving the time synchronization problem between a wearable camera and a chest-worn accelerometer sensor, the algorithm design can be further adapted to other sensing modalities on different parts of the human body. Doing so will enable temporal alignment of video-derived labels to diverse wearable sensor signals. Such temporally precise labels obtained from the in-the-wild environment can improve the accuracy of detecting fine-grained micro-behaviors (e.g., dynamics of hand-to-mouth gestures) underlying a wide variety of daily behaviors such as eating, drinking, brushing, flossing, and smoking. In addition to advancing computational models for detecting daily behaviors, fine-grained observations of micro-behaviors in the wild can also advance our understanding of daily human behaviors. Finally, we provide our novel S2S-Sync dataset to the research community, for use as a benchmark for in-the-wild time synchronization, along with our time synchronization software, including scripts for reproducing all of the experiments in this manuscript, so as to continue to advance this area of research.

CHAPTER 6
CONCLUSIONS AND FUTURE WORK

Wearable sensors are increasingly being used in healthcare research and consumer products. Wearable cameras provide a unique sensing modality for analyzing health-related behaviors. Body-worn cameras capture the context and the actions of a participant in their daily life environment, and this dissertation demonstrates three ways in which egocentric cues can be utilized for health-related applications. First, my work on zero-shot action recognition in Chapter 3 demonstrates that egocentric video can be analyzed to detect cooking actions defined as noun-verb combinations (*e.g.*, spread peanut butter) and that models for novel noun-verb pairs can be constructed without the need to collect additional training data. Second, my work on the automatic detection of screen-watching events in Chapter 4 demonstrates that the applications of egocentric video analysis extend beyond the more widely studied tasks of cooking, eating, and drinking detection, and can encompass the measurement of attention to screens (and potentially to other targets in the visual environment, such as billboards and advertising, in the future). Thus, wearable cameras without eye-tracking capabilities can be used as a sensor for visual exposure and attention. This finding is significant given the much higher cost and complexity of eye-tracking systems in comparison to video recorders. Third, my work on the automatic synchronization of egocentric video with accelerometry in Chapter 5 demonstrates that video captured in the naturalistic environment can be time-synchronized with other wearable sensing modalities in a scalable manner that does not pose a significant burden for participants. This finding is salient given the increasing feasibility of deploying wearable cameras in a mobile health study and the value of these cameras in providing ground truth information about daily life activities that can inform the development of machine learning models for accelerometry and other on-body sensors. In conducting this research, multiple datasets of egocentric

video were collected in naturalistic settings, including in the Aware Home Lab at Georgia Tech and in an in-the-wild study in which participants in the Atlanta area captured egocentric videos during their daily life activities.

6.1 Future Work

As a means to demonstrate how the technologies we have created in this dissertation could be integrated and extended to impact health outcomes, we now describe a particular scenario that is one of the directions of future work for this project. This scenario is based on the research activities of Dr. David Conroy's lab at Pennsylvania State University and results from a recent collaboration. One important health-related dimension of screen use is the correlation between extended periods of screen use and unhealthy behaviors, such as physical inactivity, smoking, mindless eating and snacking. One explanation for extended screen use (*e.g.*, binge-watching Netflix) is that screen content is designed to be engaging and entertaining, creating a positive feedback loop in which exposure to screen content prompts a desire to consume more content. This phenomenon could make it more difficult to incorporate a modest amount of screen watching into a health daily routine that includes significant physical activity, well-defined meal times, and other practices that promote positive health outcomes.

From this perspective, a key question is to identify the triggers that result in the initiation of screen watching, which can precipitate extended viewing sessions and contribute to the exclusion of other more healthy activities. Such identification could lead to a greater understanding of how screen use could be more effectively managed. Dr. Conroy's team made a step towards this goal by collecting a multimodal TV-watching dataset. They recruited subjects with significantly high amounts of screen use and collected data from participants at home to gain insight into their screen-use behavior. Subjects were required to wear a Pivothead wearable camera and an accelerometer attached to the thigh, making it possible to quantify TV-watching behavior (via analysis of the collected video) and detect moments

of physical activity and inactivity via accelerometry, where moments were categorized as standing, walking, or sitting. This dataset could be used to answer the following research questions:

1. What contextual cues (*e.g.*, sofa, food, snack) are frequently associated with the onset of extended screen use?

2. Can these contextual cues be viewed as triggers that precipitate screen use?

3. What is the frequency of co-occurring behaviors associated with screen viewing? (*e.g.*, phone use, eating)

4. What contextual cues are frequently associated with the onset of sitting and other sedentary behaviors independent of screen use?

The methods from this dissertation could be extended and utilized to address these research questions. The methods from Chapter 4 can be used to quantify exposure to screens and identify the moments in time that precede an extended bout of screen use. Analysis of the video preceding the onset of watching, performed by both researchers and algorithms, could then be used to identify potential predictors of screen use. Similarly, analysis of accelerometry data can be used to quantify sitting behaviors and identify the moments that precede sedentary activity. The methods from Chapter 5 can be used to automatically synchronize video and accelerometry signals and identify cues in both modalities. Moreover, if effective predictors of extended screen use or sedentary activity were identified, the methods from this dissertation would provide a starting point for developing automated methods for detecting the presence of such cues, as a step towards developing a mobile intervention that could support more effective management of screen time as part of a healthy daily routine. Furthermore, our synchronization method could be extended to address other sensing modalities, such as ballistocardiogram (BCG), respiratory inductance plethysmography (RIP) sensor, which could, in turn, increase the opportunity to identify novel cues and develop behavioral biomarkers.

6.1.1 Joint Learning of Synchronization and Action Recognition

This dissertation presents a synchronization method for video and accelerometer that is based on classical machine learning and statistical techniques. An exciting dimension for future work is to investigate multitask learning with deep neural network models to jointly estimate the synchronization offset and learn meaningful representations for action recognition. The idea is that the synchronization between data streams of different modalities can provide a self-supervisory signal to enable deep neural networks to learn effective representations. The learned representations can be further fine-tuned for other tasks. For example, a representation that is trained to identify the time alignment between video and accelerometry is a potentially good starting point for learning to recognize activities from a combination of video and accelerometry signals.

REFERENCES

[1] J. M. McGinnis and W. H. Foege, "Actual causes of death in the united states," *Jama*, vol. 270, no. 18, pp. 2207–2212, 1993.

[2] S. Babb, "Quitting smoking among adults—united states, 2000–2015," *MMWR. Morbidity and mortality weekly report*, vol. 65, 2017.

[3] T. M. Bush, M. D. Levine, B. Magnusson, Y. Cheng, X. Chen, L. Mahoney, L. Miles, and S. M. Zbikowski, "Impact of baseline weight on smoking cessation and weight gain in quitlines," *Annals of Behavioral Medicine*, vol. 47, no. 2, pp. 208–217, 2013.

[4] P. X. Nguyen, G. Rogez, C. Fowlkes, and D. Ramanan, "The open world of micro-videos," *arXiv preprint arXiv:1603.09439*, 2016.

[5] E. Thomaz, A. Parnami, J. Bidwell, I. Essa, and G. D. Abowd, "Technological approaches for addressing privacy concerns when recognizing eating behaviors with wearable cameras," in *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '13, Zurich, Switzerland: Association for Computing Machinery, 2013, 739–748, ISBN: 9781450317702.

[6] B. Bell, R. Alam, N. Alshurafa, E. Thomaz, A. Mondol, K. de la Haye, J. Stankovic, J. Lach, and D. Spruijt-Metz, "Automatic, wearable-based, in-field eating detection approaches for public health research: A scoping review," *npj Digital Medicine*, vol. 3, no. 1, Dec. 2020, Publisher Copyright: © 2020, The Author(s). Copyright: Copyright 2020 Elsevier B.V., All rights reserved.

[7] Y. C. Zhang, Y. Li, and J. M. Rehg, "First-person action decomposition and zero-shot learning," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2017, pp. 121–129.

[8] Y. C. Zhang and J. M. Rehg, "Watching the tv watchers," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, no. 2, Jul. 2018.

[9] Y. C. Zhang, S. Zhang, M. Liu, E. Daly, S. Battalio, S. Kumar, B. Spring, J. M. Rehg, and N. Alshurafa, "Syncwise: Window induced shift estimation for synchronization of video and accelerometry from wearable sensors," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 4, no. 3, Sep. 2020.

[10] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," in *CVPR*, 2009, pp. 2929–2936.

[11] T.-H. Vu, C. Olsson, I. Laptev, A. Oliva, and J. Sivic, "Predicting actions from static scenes," in *ECCV*, 2014, pp. 421–436.

[12] A. Gupta, A. Kembhavi, and L. S. Davis, "Observing human-object interactions: Using spatial and functional compatibility for recognition," *Trans. PAMI*, vol. 31, no. 10, pp. 1775–1789, 2009.

[13] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. M. Rehg, "A scalable approach to activity recognition based on object use," in *ICCV*, 2007, pp. 1–8.

[14] B. Yao and L. Fei-Fei, "Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses," *Trans. PAMI*, vol. 34, no. 9, pp. 1691–1703, 2012.

[15] M. Wray, D. Moltisanti, W. Mayol-Cuevas, and D. Damen, "Sembed: Semantic embedding of egocentric action videos," in *ECCV Workshop on Egocentric Perception, Interaction and Computing*, 2016, pp. 532–545.

[16] K. Matsuo, K. Yamada, S. Ueno, and S. Naito, "An attention-based activity recognition for egocentric video," in *CVPR*, 2014, pp. 551–556.

[17] T. McCandless and K. Grauman, "Object-centric spatio-temporal pyramids for egocentric activity recognition.," in *BMVC*, vol. 2, 2013, p. 3.

[18] M. Ma, H. Fan, and K. M. Kitani, "Going deeper into first-person activity recognition," in *CVPR*, 2016, pp. 1894–1903.

[19] A. Fathi and J. M. Rehg, "Modeling actions through state changes," in *CVPR*, 2013, pp. 2579–2586.

[20] A. Fathi, Y. Li, and J. M. Rehg, "Learning to recognize daily actions using gaze," in *ECCV*, 2012, pp. 314–327.

[21] H. Pirsiavash and D. Ramanan, "Detecting activities of daily living in first-person camera views," in *CVPR*, 2012, pp. 2847–2854.

[22] H. S. Park and J. Shi, "Social saliency prediction," in *CVPR*, 2015, pp. 4777–4785.

[23] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng, "Zero-shot learning through cross-modal transfer," in *NIPS*, 2013, pp. 935–943.

[24] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *CVPR*, 2009, pp. 951–958.

[25] O. Russakovsky and L. Fei-Fei, "Attribute learning in large-scale datasets," in *ECCV*, 2010, pp. 1–14.

[26] S. Antol, C. L. Zitnick, and D. Parikh, "Zero-shot learning via visual abstraction," in *ECCV*, 2014, pp. 401–416.

[27] M. Liu, S. Chen, and D. Zhang, "Learning attribute relation in attribute-based zero-shot classification," in *International Conference on Intelligent Science and Intelligent Data Engineering*, 2012, pp. 514–521.

[28] Z. Zhang and V. Saligrama, "Zero-shot learning via semantic similarity embedding," in *ICCV*, 2015, pp. 4166–4174.

[29] J. Liu, B. Kuipers, and S. Savarese, "Recognizing human actions by attributes," in *CVPR*, 2011, pp. 3337–3344.

[30] H.-T. Cheng, F.-T. Sun, M. Griss, P. Davis, J. Li, and D. You, "Nuactiv: Recognizing unseen new activities using semantic attribute-based learning," in *International Conference on Mobile Systems, Applications, and Services*, 2013, pp. 361–374.

[31] X. Xu, T. M. Hospedales, and S. Gong, "Semantic embedding space for zero-shot action recognition," in *ICIP*, 2015.

[32] Q. Wang and K. Chen, "Zero-shot visual recognition via bidirectional latent embedding," *arXiv preprint arXiv:1607.02104*, 2016.

[33] M. Jain, J. C. van Gemert, T. Mensink, and C. G. Snoek, "Objects2action: Classifying and localizing actions without any video example," in *ICCV*, 2015, pp. 4588–4596.

[34] R. Navarathna, P. Lucey, P. Carr, E. Carter, S. Sridharan, and I. Matthews, "Predicting movie ratings from audience behaviors," in *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, IEEE, 2014, pp. 1058–1065.

[35] J. Hernandez, Z. Liu, G. Hulten, D. DeBarr, K. Krum, and Z. Zhang, "Measuring the engagement level of tv viewers," in *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, 2013, pp. 1–7.

[36] D. Lee, W. han Yun, C. kyu Park, H Yoon, J. Kim, and C. Park, "Measuring the engagement level of children for multiple intelligence test using kinect," in *Seventh International Conference on Machine Vision (ICMV 2014)*, International Society for Optics and Photonics, vol. 9445, 2015, p. 944 529.

[37] M. Takahashi, S. Clippingdale, M. Naemura, and M. Shibata, "Estimation of viewers' ratings of tv programs based on behaviors in home environments," *Multimedia Tools and Applications*, vol. 74, no. 19, pp. 8669–8684, 2015.

[38] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba, "Eye tracking for everyone," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[39] Q. Huang, A. Veeraraghavan, and A. Sabharwal, "Tabletgaze: Unconstrained appearance-based gaze estimation in mobile tablets," *arXiv preprint arXiv:1508.01244*, 2015.

[40] Z. Ye, Y. Li, Y. Liu, C. Bridges, A. Rozga, and J. M. Rehg, "Detecting bids for eye contact using a wearable camera," in *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, IEEE, vol. 1, 2015, pp. 1–8.

[41] E. Chong, K. Chanda, Z. Ye, A. Southerland, N. Ruiz, R. M. Jones, A. Rozga, and J. M. Rehg, "Detecting gaze towards eyes in natural social interactions and its use in child assessment," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 3, 43:1–43:20, Sep. 2017.

[42] B. A. Smith, Q. Yin, S. K. Feiner, and S. K. Nayar, "Gaze locking: Passive eye contact detection for human-object interaction," in *Proceedings of the 26th annual ACM symposium on User interface software and technology*, ACM, 2013, pp. 271–280.

[43] R. Fletcher, D Chamberlain, D Richman, N Oreskovic, and E Taveras, "Wearable sensor and algorithm for automated measurement of screen time," pp. 1–8, 2016.

[44] J. Kallenbach, S. Narhi, and P. Oittinen, "Effects of extra information on tv viewers' visual attention, message processing ability, and cognitive workload," *Computers in Entertainment (CIE)*, vol. 5, no. 2, p. 8, 2007.

[45] R.-D. Vatavu and M. Mancas, "Evaluating visual attention for multi-screen television: Measures, toolkit, and experimental findings," *Personal and Ubiquitous Computing*, vol. 19, no. 5-6, pp. 781–801, 2015.

[46] J. R. Cauchard, M. Löchtefeld, P. Irani, J. Schoening, A. Krüger, M. Fraser, and S. Subramanian, "Visual separation in mobile multi-display environments," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, ACM, 2011, pp. 451–460.

[47] A. Apaolaza, A. Brown, C. Jay, and S. Harper, *Understanding the division of attention between TV and companion content: experiment 2, without eye-tracking*. Technical report, Oct. 2014.

[48] A. Apaolaza, R. Haines, A. Aizpurua, A. Brown, M. Evans, S. Jolly, S. Harper, and C. Jay, "Abc: Using object tracking to automate behavioural coding," in *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, ACM, 2016, pp. 2766–2773.

[49] A. Loveday, L. B. Sherar, J. P. Sanders, P. W. Sanderson, and D. W. Esliger, "Novel technology to help understand the context of physical activity and sedentary behaviour," *Physiological Measurement*, vol. 37, no. 10, p. 1834, 2016.

[50] J. Kerr, S. J. Marshall, S. Godbole, J. Chen, A. Legge, A. R. Doherty, P. Kelly, M. Oliver, H. M. Badland, and C. Foster, "Using the sensecam to improve classifications of sedentary behavior in free-living settings," *American journal of preventive medicine*, vol. 44, no. 3, pp. 290–296, 2013.

[51] B. E. Ainsworth, W. L. Haskell, S. D. Herrmann, N. Meckes, D. R. Bassett Jr, C. Tudor-Locke, J. L. Greer, J. Vezina, M. C. Whitt-Glover, and A. S. Leon, "2011 compendium of physical activities: A second update of codes and met values.," *Medicine and science in sports and exercise*, vol. 43, no. 8, pp. 1575–1581, 2011.

[52] M. Hayhoe and D. Ballard, "Eye movements in natural behavior," *Trends in cognitive sciences*, vol. 9, no. 4, pp. 188–194, 2005.

[53] S. E. Petersen and M. I. Posner, "The attention system of the human brain: 20 years after," *Annual review of neuroscience*, vol. 35, pp. 73–89, 2012.

[54] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[55] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille, "The Secrets of Salient Object Segmentation," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 280–287.

[56] S. Chaabouni, J. Benois-Pineau, and C. B. Amar, "Transfer learning with deep networks for saliency prediction in natural video," in *Image Processing (ICIP), 2016 IEEE International Conference on*, IEEE, 2016, pp. 1604–1608.

[57] M. Land, N. Mennie, and J. Rusted, "The roles of vision and eye movements in the control of activities of daily living," *Perception*, vol. 28, pp. 1311–1328, 1999.

[58] B. W. Tatler, "The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions," *Journal of Vision*, vol. 7, no. 14, pp. 1–17, 2007.

[59] F. Cristino and R. Baddeley, "The nature of the visual representations involved in eye movements when walking down the street," *Visual Cognition*, vol. 17, pp. 880–903, 2009.

[60] B. Tatler, M. Hayhoe, M. Land, and D. Ballard, "Eye guidance in natural vision: Reinterpreting salience," *Journal of Vision*, vol. 11, no. 5, pp. 1–23, 2011.

[61] Y. Li, A. Fathi, and J. M. Rehg, "Learning to predict gaze in egocentric video," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3216–3223.

[62] S. Chen, J. S. Brantley, T. Kim, and J. Lach, "Characterizing and minimizing synchronization and calibration errors in inertial body sensor networks," in *Proceedings of the Fifth International Conference on Body Area Networks*, ser. BodyNets '10, Corfu, Greece, 2010, 138–144.

[63] L. Fridman, D. E. Brown, W. Angell, I. Abdić, B. Reimer, and H. Y. Noh, "Automated synchronization of driving data using vibration and steering events," *Pattern Recognition Letters*, vol. 75, pp. 9–15, 2016.

[64] P. Nilsson John-Olof; Händel, "Time synchronization and temporal ordering of asynchronous sensor measurements of a multi-sensor navigation system," in *IEEE/ION Position, Location and Navigation Symposium*, 2010, pp. 897–902.

[65] I. Skog and P. Handel, "Time synchronization errors in loosely coupled gps-aided inertial navigation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1014–1023, 2011.

[66] O. Wang, C. Schroers, H. Zimmer, M. Gross, and A. Sorkine-Hornung, "Videosnapping: Interactive synchronization of multiple videos," *ACM Trans. Graph.*, vol. 33, no. 4, Jul. 2014.

[67] P. Sand and S. Teller, "Video matching," in *ACM SIGGRAPH 2004 Papers*, ser. SIGGRAPH '04, Los Angeles, California: Association for Computing Machinery, 2004, 592–599, ISBN: 9781450378239.

[68] P. Wieschollek, I. Freeman, and H. P. Lensch, "Learning robust video synchronization without annotations," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, IEEE, 2017, pp. 92–100.

[69] J. S. Chung and A. Zisserman, "Out of time: Automated lip sync in the wild," in *Asian conference on computer vision*, Springer, 2016, pp. 251–263.

[70] Y. Aytar, C. Vondrick, and A. Torralba, "Soundnet: Learning sound representations from unlabeled video," in *Advances in neural information processing systems*, 2016, pp. 892–900.

[71] D. Harwath, A. Torralba, and J. Glass, "Unsupervised learning of spoken language with visual context," in *Advances in Neural Information Processing Systems*, 2016, pp. 1858–1866.

[72] X. Yang, P. Ramesh, R. Chitta, S. Madhvanath, E. A. Bernal, and J. Luo, "Deep multimodal representation learning from temporal data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5447–5455.

[73] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: A survey," *IEEE Network*, vol. 18, no. 4, pp. 45–50, 2004.

[74] S. M. Lasassmeh and J. M. Conrad, "Time synchronization in wireless sensor networks: A survey," in *Proceedings of the IEEE SoutheastCon 2010 (SoutheastCon)*, 2010, pp. 242–245.

[75] S. M. Hossain, T. Hnat, N. Saleheen, N. J. Nasrin, J. Noor, B.-J. Ho, T. Condie, M. Srivastava, and S. Kumar, "Mcerebrum: A mobile sensing software platform for development and validation of digital biomarkers and interventions," in *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, ser. SenSys '17, Delft, Netherlands: Association for Computing Machinery, 2017, ISBN: 9781450354592.

[76] S. Akther, N. Saleheen, S. A. Samiei, V. Shetty, E. Ertin, and S. Kumar, "Moral: An mhealth model for inferring oral hygiene behaviors in-the-wild using wrist-worn inertial sensors," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 3, no. 1, Mar. 2019.

[77] F. de la Torre, J. K. Hodgins, J. Montano, and S. Valcarcel, "Detailed human data acquisition of kitchen activities: The cmu-multimodal activity database (cmu-mmac)," in *CHI 2009 Workshop. Developing Shared Home Behavior Datasets to Advance HCI and Ubiquitous Computing Research*, 2009.

[78] J.-H. Bae and J.-T. Kim, "Design and implementation of high accurate synchronization between gyroscope and image sensor," in *2015 Seventh International Conference on Ubiquitous and Future Networks*, IEEE, 2015, pp. 956–958.

[79] E. Cippitelli, S. Gasparrini, E. Gambi, S. Spinsante, J. Wåhslény, I. Orhany, and T. Lindhy, "Time synchronization and data fusion for rgb-depth cameras and inertial sensors in aal applications," in *2015 IEEE International Conference on Communication Workshop (ICCW)*, IEEE, 2015, pp. 265–270.

[80] T. Plotz, C. Chen, N. Y. Hammerla, and G. D. Abowd, "Automatic synchronization of wearable sensors and video-cameras for ground truth annotation – a practical approach," in *2012 16th International Symposium on Wearable Computers*, 2012, pp. 100–103.

[81] Y. C. Han, K. I. Wong, and I. Murray, "Automatic synchronization of markerless video and wearable sensors for walking assessment," *IEEE Sensors Journal*, 2019.

[82] D. Bannach, O. Amft, and P. Lukowicz, "Automatic event-based synchronization of multimodal data streams from wearable and ambient sensors," *European Conference on Smart Sensing and Context*, vol. 5741, pp. 135–148, 2009.

[83] H. Brugman and A. Russel, "Annotating multi-media/multi-modal resources with ELAN," in *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal: European Language Resources Association (ELRA), May 2004.

[84] A. Fouse, N. Weibel, E. Hutchins, and J. Hollan, "Chronoviz: A system for supporting navigation of time-coded data," Jan. 2011, pp. 299–304.

[85] R. Alharbi, A. Pfammatter, B. Spring, and N. Alshurafa, "Willsense: Adherence barriers for passive sensing systems that track eating behavior," in *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '17, Denver, Colorado, USA: Association for Computing Machinery, 2017, 2329–2336, ISBN: 9781450346566.

[86] R. Alharbi, T. Stump, N. Vafaie, A. Pfammatter, B. Spring, and N. Alshurafa, "I can't be myself: Effects of wearable cameras on the capture of authentic behavior in the wild," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, no. 3, Sep. 2018.

[87] S. Zhang, Y. Zhao, D. T. Nguyen, R. Xu, S. Sen, J. Hester, and N. Alshurafa, "Necksense: A multi-sensor necklace for detecting eating activities in free-living conditions," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 4, no. 2, Jun. 2020.

[88] J. W. Kamminga, L. M. Janßen, N. Meratnia, and P. J. M. Havinga, "Horsing around—a dataset comprising horse movement," *Data*, vol. 4, no. 4, 2019.

[89] V. Radu and M. Henne, "Vision2sensor: Knowledge transfer across sensing modalities for human activity recognition," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 3, no. 3, Sep. 2019.

[90] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei, "Human action recognition by learning bases of action attributes and parts," in *ICCV*, 2011, pp. 1331–1338.

[91] T. S. Motwani and R. J. Mooney, "Improving video activity recognition using object recognition and text mining," in *European Conference on Artificial Intelligence*, 2012, pp. 600–605.

[92] Y. Li, Z. Ye, and J. M. Rehg, "Delving into egocentric actions," in *CVPR*, 2015.

[93] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *ICCV*, 2013, pp. 3551–3558.

[94] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *CVPR*, 2014, pp. 1725–1732.

[95] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *ECCV*, 2010, pp. 143–156.

[96] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition.," in *ICML*, vol. 32, 2014, pp. 647–655.

[97] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014, pp. 580–587.

[98] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

[99] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *CVPR Workshop on DeepVision*, 2014, pp. 806–813.

[100] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *Trans. PAMI*, vol. 35, no. 1, pp. 221–231, 2013.

[101] Y. Poleg, A. Ephrat, S. Peleg, and C. Arora, "Compact CNN for indexing egocentric videos," in *WACV*, 2016, pp. 1–9.

[102] L. Ladicky, C. Russell, P. Kohli, and P. H. Torr, "Graph cut based inference with co-occurrence statistics," in *ECCV*, 2010, pp. 239–253.

[103] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, 27:1–27:27, 3 2011, Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[104] Y. Li, A. Fathi, and J. M. Rehg, "Learning to predict gaze in egocentric video," in *ICCV*, 2013, pp. 3216–3223.

[105] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *ECCV*, 2014, pp. 391–405.

[106] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *The Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[107] Y. Dong, S. Gao, K. Tao, J. Liu, and H. Wang, "Performance evaluation of early and late fusion methods for generic semantics indexing," *Pattern Analysis and Applications*, vol. 17, no. 1, pp. 37–50, 2014.

[108] C. G. Snoek, M. Worring, and A. W. Smeulders, "Early versus late fusion in semantic video analysis," in *ACM International Conference on Multimedia*, 2005, pp. 399–402.

[109] U. B. of Labor Statistics. (2016). "Atus table. time spent in detailed primary activities and percent of the civilian population engaging in each activity, averages per day by sex, annual averages," (visited on 04/21/2018).

[110] W. H. Dietz and S. L. Gortmaker, "Do we fatten our children at the television set? obesity and television viewing in children and adolescents," *Pediatrics*, vol. 75,

no. 5, pp. 807–812, 1985. eprint: http://pediatrics.aappublications.org/content/75/5/807.full.pdf.

[111] J. G. Johnson, P. Cohen, S. Kasen, and J. S. Brook, "Extensive television viewing and the development of attention and learning difficulties during adolescence," *Archives of pediatrics & adolescent medicine*, vol. 161, no. 5, pp. 480–486, 2007.

[112] J. G. Johnson, P. Cohen, E. M. Smailes, S. Kasen, and J. S. Brook, "Television viewing and aggressive behavior during adolescence and adulthood," *Science*, vol. 295, no. 5564, pp. 2468–2471, 2002.

[113] D. P. Hallahan, J. M. Kauffman, and P. C. Pullen, *Exceptional learners: An introduction to special education*. Pearson Higher Ed, 2011.

[114] E. L. Swing, D. A. Gentile, C. A. Anderson, and D. A. Walsh, "Television and video game exposure and the development of attention problems," *Pediatrics*, vol. 126, no. 2, pp. 214–221, 2010.

[115] C. E. Landhuis, R. Poulton, D. Welch, and R. J. Hancox, "Does childhood television viewing lead to attention problems in adolescence? results from a prospective longitudinal study," *Pediatrics*, vol. 120, no. 3, pp. 532–537, 2007.

[116] F. J. Zimmerman and D. A. Christakis, "Children's television viewing and cognitive outcomes: A longitudinal analysis of national data," *Archives of Pediatrics & Adolescent Medicine*, vol. 159, no. 7, pp. 619–625, 2005.

[117] H. Takeuchi, Y. Taki, H. Hashizume, K. Asano, M. Asano, Y. Sassa, S. Yokota, Y. Kotozaki, R. Nouchi, and R. Kawashima, "The impact of television viewing on brain structures: Cross-sectional and longitudinal analyses," *Cerebral Cortex*, bht315, 2013.

[118] T. Northup, "Understanding the relationship between television use and unhealthy eating: The mediating role of fatalistic views of eating well and nutritional knowledge," *The International Journal of Communication and Health*, vol. 1, no. 3, pp. 10–15, 2014.

[119] J. L. Harris and J. A. Bargh, "Television viewing and unhealthy diet: Implications for children and media interventions," *Health communication*, vol. 24, no. 7, pp. 660–673, 2009.

[120] R. Mejia, A. Perez, E. N. Abad-Vivero, C. Kollath-Cattano, I. Barrientos-Gutierrez, J. F. Thrasher, and J. D. Sargent, "Exposure to alcohol use in motion pictures and teen drinking in latin america," *Alcoholism: Clinical and Experimental Research*, 2016.

[121] M. Morgenstern, Z. Li, Z. Li, and J. D. Sargent, "The party effect: Prediction of future alcohol use based on exposure to specific alcohol advertising content," *Addiction*, 2016.

[122] A. N. M. Research. (). "Unitam," (visited on 04/21/2018).

[123] R. van Brandenburg, H. van den Berg, M. O. van Deventer, and I. M. Schenk, "Towards multi-user personalized tv services, introducing combined rfid digest authentication," 2009.

[124] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[125] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *Int. J. Comput. Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[126] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, Springer, 2014, pp. 740–755.

[127] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," *arXiv:1602.00763*, 2016.

[128] J. Shi and C. Tomasi, "Good features to track," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, 1994, pp. 593–600.

[129] J. yves Bouguet, "Pyramidal implementation of the lucas kanade feature tracker," *Intel Corporation, Microprocessor Research Labs*, 2000.

[130] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*, ser. Adaptive computation and machine learning. MIT Press, 2009, ISBN: 9780262013192.

[131] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*, Springer, 2016, pp. 21–37.

[132] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *European Conference on Computer Vision*, Springer, 2016, pp. 525–542.

[133] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[134] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[135] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," *arXiv preprint arXiv:1707.01083*, 2017.

[136] A. Fathi and J. M. Rehg, "Modeling actions through state changes," in *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '13, Washington, DC, USA: IEEE Computer Society, 2013, pp. 2579–2586, ISBN: 978-0-7695-4989-7.

[137] E. H. Spriggs, F. De La Torre, and M. Hebert, "Temporal segmentation and activity classification from first-person sensing," in *2009 IEEE Computer Society Confer-*

ence on Computer Vision and Pattern Recognition Workshops, IEEE, 2009, pp. 17–24.

[138] J. M. Rehg, S. Murphy, and S. Kumar, *Mobile Health: Sensors, Analytic Methods, and Applications*. Springer, 2017.

[139] N. Alshurafa, A. W. Lin, F. Zhu, R. Ghaffari, J. Hester, E. Delp, J. Rogers, and B. Spring, "Counting bites with bits: Expert workshop addressing calorie and macronutrient intake monitoring," *J Med Internet Res*, vol. 21, no. 12, e14904, 2019.

[140] S. Bi, T. Wang, N. Tobias, J. Nordrum, S. Wang, G. Halvorsen, S. Sen, R. Peterson, K. Odame, K. Caine, and et al., "Auracle: Detecting eating episodes with an ear-mounted sensor," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, no. 3, Sep. 2018.

[141] T. S. Limited, *Timecode systems*, `https://www.timecodesystems.com/syncbac-pro/`, Accessed: 2020-05-15, 2020.

[142] I. Freeman, P. Wieschollek, and H. P. A. Lensch, "Robust video synchronization using unsupervised deep learning," *CoRR*, vol. abs/1610.05985, 2016. arXiv: `1610.05985`.

[143] *Amazing grace wikipedia website*, `https://en.wikipedia.org/wiki/Amazing_Grace_(2018_film)`, Retrieved February, 2020.

[144] B. M. Booth, K. Mundnich, T. Feng, A. Nadarajan, T. H. Falk, J. L. Villatte, E. Ferrara, and S. Narayanan, "Multimodal human and environmental sensing for longitudinal behavioral studies in naturalistic settings: Framework for sensor selection, deployment, and management," *J Med Internet Res*, vol. 21, no. 8, e12832, 2019.

[145] E. Z. Welty, T. C. Bartholomaus, S. O'Neel, and W. Tad Pfeffer, "Cameras as clocks," *Journal of Glaciology*, vol. 59, no. 214, 275–286, 2013.

[146] E. Ertin, N. Stohs, S. Kumar, A. Raij, M. al'Absi, and S. Shah, "Autosense: Unobtrusively wearable sensor suite for inferring the onset, causality, and consequences of stress in the field," in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '11, Seattle, Washington: Association for Computing Machinery, 2011, 274–287, ISBN: 9781450307185.

[147] MD2K, *Motionsense - md2k*, `https://md2k.org/documentation/data_dictionary/raw_streams/motionsense.html`, Accessed: 2020-05-15, 2020.

[148] T. Hnat, S. Hossain, N. Ali, S. Carini, T. Condie, I. Sim, M. Srivastava, and S. Kumar, "Mcerebrum and cerebral cortex: A real-time collection, analytic, and intervention platform for high-frequency mobile sensor data," in *AMIA (American Medical Informatics Association) 2017 Annual Symposium*, American Medical Informatics Association, 2017, published.

[149] N. Saleheen, A. A. Ali, S. M. Hossain, H. Sarker, S. Chatterjee, B. Marlin, E. Ertin, M. al'Absi, and S. Kumar, "Puffmarker: A multi-sensor approach for pinpointing the timing of first lapse in smoking cessation," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. Ubi-

Comp '15, Osaka, Japan: Association for Computing Machinery, 2015, 999–1010, ISBN: 9781450335744.

[150] K. Hovsepian, M. al'Absi, E. Ertin, T. Kamarck, M. Nakajima, and S. Kumar, "Cstress: Towards a gold standard for continuous stress assessment in the mobile environment," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '15, Osaka, Japan: Association for Computing Machinery, 2015, 493–504, ISBN: 9781450335744.

[151] M. Liu, X. Chen, Y. Zhang, Y. Li, and J. M. Rehg, "Attention distillation for learning video representations," in *BMVC*, 2020.

[152] Y. Li, M. Liu, and J. M. Rehg, "In the eye of beholder: Joint learning of gaze and actions in first person video," in *ECCV*, 2018.

[153] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8934–8943.

[154] T. S. community, *Scipy library curve fitting method*, `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html`, Accessed: 2020-05-15, 2019.

[155] A. A. Amin, S. M. Hossain, K. Hossain, M. Hovsepian, M. Rahman, K. Plarre, and S. Kumar, "Mpuff: Automated detection of cigarette smoking puffs from respiration measurements," in *2012 ACM/IEEE 11th International Conference on Information Processing in Sensor Networks (IPSN)*, 2012, pp. 269–280.